



User's Manual for

JetProg

Universal 48-pindrive Programmer,
expandable up to 256.

LabProg+

Universal 48-pindrive Programmer

PREPROM-02aLV

Universal Memory Programmer

51&AVRprog

MCS51 Series and Atmel AVR Microcontrollers
Programmer

PIKprog

Microchip PICmicro Programmer

SEEplog

Serial EEPROM Programmer

ELNEC s.r.o.
Presov, Slovakia
February 2001

LabProg+ Best.-Nr. 15 00 53

Programming Times

Note: The programming times of the programmer LabProg+, attached to PC through LPT port, depends a bit on PC speed, LPT port type and operating system free resources. Look at www.elnec.com - [ELNEC programmers, tips to increase of programming speed](#) application note for details. If you plan to use high-capacity memories, please accept a long programming time or think about using much more powerful [BeeProg Best.-Nr. 15 01 70](#) or [JetProg Best.-Nr. 15 00 21](#) programmer.

For comparison of programming speed BeeProg/ LabProg+ here is the table:

Device	Operation	Time BeeProg	Time LabProg+
AT29C040A	programming and verify	26 sec.	75 sec.
AM29F040	programming and verify	28 sec.	165 sec.
PIC16C67	programming and verify	6 sec.	30 sec.



**COPYRIGHT © 1997 - 2001
ELNEC s.r.o.**

This document is copyrighted by ELNEC s.r.o, Presov, Slovakia. All rights reserved. This document or any part of it may not be copied, reproduced or translated in any form or in any way without the prior written permission of ELNEC s.r.o.

The control program is copyright ELNEC s.r.o., Presov, Slovakia. The control program or any part of it may not be analyzed, disassembled or modified in any form, on any medium, for any purpose.

Information provided in this manual is intended to be accurate at the moment of release, but we continuously improve all our products. Please consult the README_P.TXT on the enclosed floppy disk for last minute changes.

ELNEC s.r.o. assumes no responsibility for misuse of this manual.

ELNEC s.r.o. reserves the right to make changes or improvements to the product described in this manual at any time without notice. This manual contains names of companies, software products, etc., which may be trademarks of their respective owners. ELNEC s.r.o. respects those trademarks



How to use this manual

This manual explains how to install the control program and how to use your programmer. It is assumed that the user has some experience with PCs and installation of software. Once you have installed the control program we

recommend you consult the context sensitive HELP within the control program rather than the printed User's Manual. Revisions are implemented in the context sensitive help before the printed Users Manual.

Dear customer,

thank you for
purchasing one of the
ELNEC programmer.

Note: Because this user's manual is common for all ELNEC programmers, read please section(s) respective programmer you have bought.

This manual contains two main sections:

Quick Start

Read this section if you are an experienced user. You will find only specific information regarding installation of the control program and use of your programmer. For more detailed instructions you may read the **Description in detail** section or the **Troubleshooting** chapter for the respective programmer.

Descriptions in detail for each programmer

Read this section for the respective programmer if you are a less experienced user or if you need detailed information. You may find some less relevant features of programmer described here, but all programmer features are described in this section along with details regarding installation of the control program. Read this section to explore all of the features provided by your programmer.

Download please actual version of manual from
ELNEC WEB site (www.elnec.com) if current one
will be out of date.



Table of contents

How to use this manual.....	3
Introduction.....	9
Products configuration	12
PC requirements.....	13
Quick Start.....	15
JetProg	17
Introduction.....	18
JetProg elements.....	22
Connecting JetProg to the PC.....	23
Selftest and Calibration.....	25
Technical specification.....	26
LabProg+	31
Introduction.....	32
LabProg+ elements	35
Connecting LabProg+ to the PC	36
Selftest and Calibration.....	38
Technical specification.....	39
PREPROM-02aLV	43
Introduction.....	44
PREPROM-02aLV diagram	46
Connecting PREPROM-02aLV programmer to PC	47
Selftest and calibration.....	49
PREPROM-02aLV specification.....	50
51&AVRprog.....	53
Introduction.....	54
Connecting 51&AVRprog programmer to PC	55
51&AVRprog specification	57
PIKprog.....	59
Introduction.....	60
Connecting PIKprog programmer to PC.....	61
PIKprog specification	63
SEEprog.....	65
Introduction.....	66
Connecting SEEprog programmer to PC	67
SEEprog specifications.....	69
Software.....	71
The programmer software.....	72
File	76
Device	78
Buffer.....	90
Options	96
Diagnostics.....	101
Help	102
Common notes	103
Software	104
Hardware.....	105
Other	106
Troubleshooting and warranty	107
Throubleshooting.....	108



If you have an unsupported target device.....	110
Warranty terms	111
Appendix.....	113
Appendix A - Programmer Problem Report form	115
Appendix B - Device Problem Report form	117
Appendix C - Keep-Current Service	119
Appendix D - AlgOR service.....	121
Appendix E - registration card	123
Appendix F - CE certifikates.....	125



Conventions used in the manual

References to the control program functions are in bold, e.g. **Load**, **File**, **Device**, etc. References to control keys are written in brackets <>, e.g. <**F1**>.

Terminology used in the manual:

- Device** any kind of programmable integrated circuits or programmable devices
- ZIF socket** Zero Insertion Force socket used for insertion of target device
- Buffer** part of memory or disk, used for temporary data storage
- Printer port** type of port of PC (parallel), which is primarily dedicated for printer connection.
- HEX data format** - format of data file, which may be read with standard text viewers; e.g. byte 5AH is stored as characters '5' and 'A', which mean bytes 35H and 41H. One line of this HEX file (one record) contains start address, data bytes and all records are secured with checksum.



Introduction



This user's manual covers all ELNEC programmers: **JetProg**, **LabProg+**, **PREPROM-02aLV**, **SEEplog**, **51&AVRprog**, **PIKprog** and **SEEplog**.

JetProg is a universal programmer and logic IC tester with 48 powerful pindrivers in base configuration, expandable up to 256. This design allows to easily add new devices to the device list. Provides very competitive price but excellent hardware design for reliable programming. Best "value for money" in this class.

LabProg+ is a universal programmer and logic IC tester with 48 powerful pindrivers. This design allows to easily add new devices to the device list. LabProg+ is a true universal and a true low cost programmer, providing one of the best "value for money" in today's market.

PREPROM-02aLV is a small and powerful EPROM, EEPROM, Flash EPROM and serial EEPROM programmer and static RAM tester, designed for professional mobile applications. In addition, PREPROM-02aLV programmer with auxiliary modules support also microprocessors (MCS48, MCS51, PIC, AVR), GALs, etc. Programmer can work with 'true LV' device too - from 2V.

51&AVRprog is little and portable programmer for MCS51 series and Atmel AVR microcontrollers. 51&AVRprog enables also programming serial EEPROM with interface types IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx).

PIKprog is little and portable programmer for Microchip PICmicro series of microcontrollers. PIKprog enables also programming serial EEPROM with interface types IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx).

SEEplog is universal programmer of all serial EEPROM in 8 pin DIL package. SEEplog programs EEPROM with interface IIC, SPI and Microwire, and also specialty as for example digital thermometers. The programmer supports LV (3.3V) devices too.

All programmers of our works with almost any IBM compatible PC, AT or higher, portable or desktop personal computers. No special interface card is required to connect to the PC, since programmers use the parallel (printer) port.

All programmers function flawlessly on systems running DOS, Windows 3.x, Windows 95/98 and Windows NT/2000.

All programmers are driven by an **easy-to-use, control program** with pull-down menus, hot keys and online help.



There are available two SW versions - for DOS (PG4U) and for WIN95/98/NT/2000 (PG4UW). Control program is common for all the ELNEC's programmers (JetProg, LabProg+, PREPROM-02aLV, 51&AVRprog, PIKprog and SEEprog).

Advanced design, including protection circuits, original brand components and careful manufacturing allows us to provide a **one-year** (51&AVRprog, PIKprog and SEEprog) to **three-year** (JetProg, LabProg+, PREPROM-02aLV) **warranty** on parts and labor for the programmers (limited 25,000 cycle warranty on ZIF socket).

Free additional services:

- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.

Free software updates are available from our Internet address **www.elnec.com**.

We also offer the following new services in our customer support program: Keep-Current and AlgOR.

- **Keep-Current** is a service by which ELNEC ships to you the latest version of the control program for programmer and the updated user documentation. A Keep-Current service is your hassle-free guarantee that you always have access to the latest software and documentation, at minimal cost.
- **AlgOR** (Algorithm On Request) service allows you to receive from ELNEC software support for programming devices not yet available in the current device list.

Note: *We don't recommend use programmers LabProg+, PREPROM-02aLV, 51&AVRprog and PIKprog for In-circuit programming. See FAQ on site **www.elnec.com**.*



Products configuration

Before installing and using your programmer, please carefully check that your package includes all next mentioned parts. If you find any discrepancy with respective parts list and/or if any of these items are damaged, please contact your distributor immediately.

JetProg, LabProg+ and PREPROM-02aLV programmer configuration

- programmer
- cable with two 25 pin, D-type connectors for connecting the programmer to the PC
- external power supply (suitable for respective programmer)
- diagnostic POD for selftest of programmer
- cap for ZIF socket (anti-dust cover)
- this User's manual
- floppy disk(s) or CD with the control program and additional files
- forms "PROGRAMMER PROBLEM REPORT" and "DEVICE PROBLEM REPORT"
- registration card
- shipping case

51&AVRprog, PIKprog and SEEprog programmer configuration

- programmer
- cable with two D-type connectors for connecting the programmer to the PC
- external power supply (suitable for respective programmer)
- this user's manual
- floppy disk(s) or CD with the control program and additional files
- forms "PROGRAMMER PROBLEM REPORT" and "DEVICE PROBLEM REPORT"
- registration card
- shipping case



PC requirements

Minimal PC requirements

- PC AT 10 MHz (DOS version software), PC 486 (WIN version software)
- 512 KB free RAM (DOS version software), 16MB RAM (WIN version software)
- one floppy disk drive
- HDD, 5 MB free space
- operating system MS/PC-DOS version 3.2 or later, WIN 3.xx, WIN95/98/NT/2000
- one free printer port with nothing attached

Recommended PC requirements

- Pentium PC 100MHz or higher
- 32 MB free RAM
- one floppy disk drive
- HDD with minimum 10 MB free space
- operating system: MS Windows 95/98/NT/2000
- one free bi-directional printer port with nothing attached
- for JetProg free parallel (printer) port on PCI bus, IEEE 1284 compatible (ECP/EPP)

Note: *For convenience, we suggest that you use a supplementary multi I/O card to provide an additional printer port (LPT2 for example), in order to avoid sharing the same LPT port between printer and programmer.*



Quick Start



Installing programmer hardware

- switch off the PC and programmer
- connect the communication port of programmer to a printer port of PC using cable supplied.
- switch on the PC
- connect the connector of the power supply adapter to the programmer

Installing the programmer software

Run the installation program from the floppy disk or CD (DOS: Install.exe, WIN: Setup.exe) and follow the on-screen instructions. Please read the README_P.TXT file for latest information about the programmer hardware and software.

Using programmer software

Launch PG4U.EXE (DOS) or PG4UW.EXE (WIN) to enter the control program. The menu **Device** contains the device manipulation commands. The menu **File** contains commands for files and directories. The menu **Buffer** is to be used for buffer manipulation.

Programming a device - the shortest way

Use the hot key <Alt+F5> to input the device name and/or manufacturer to select the desired type of target device. If you want to copy an existing device, insert it into the ZIF socket of the programmer and then press key <F7>. If you want to program a target device with data from a disk press key <F3> and read the appropriate file into the buffer. Then insert your target device into the ZIF socket. To check if the device is blank - press key <F6>. Now you can program the device by pressing key <F9>. After programming you may perform additional verification by pressing key <F8>.



JetProg



Introduction

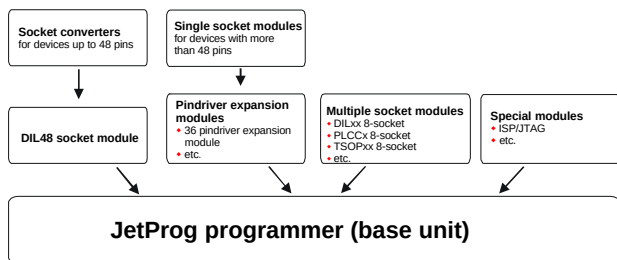
JetProg is a new generation of Windows 95/98/ME/NT/2000 based ELNEC universal programmers built to meet the rigorous demands of the leading engineers and programming centers.

JetProg supports all kinds of types and silicon technologies of programmable devices. It provides very competitive price but excellent hardware design for reliable programming. Best "value for money" in this class.

JetProg interfaces with the IBM PC, AT or higher, portable or desktop personal computers. Programmer allows you to directly connect to your PC through **any standard parallel (printer) port** (no special interface card needed). We recommend to use **parallel (printer) port on PCI bus, IEEE 1284 compatible (ECP/EPP)**. The JetProg control program support standard IEEE1284 also.

JetProg offer very fast programming due high-speed FPGA driven hardware and support of ECP/EPP parallel port. Consequently and due special protocol is communication between PC and JetProg programmer fast and very reliable. The programming AT29C040A takes about 28 seconds it is faster than most its competitors. As a result, this programmer is optional solution for middle quantities programming in production or programming centers.

Scheme of JetProg programming system



JetProg, base configuration

- JetProg, base unit
- JetProg, DIL48 socket module

For following text, term JetProg means JetProg in base configuration.



JetProg has 48 powerful pindrivers in base unit, expandable up to 256 pindrivers using "pindriver expansion" modules. Advanced pin drivers incorporate high-quality high-speed circuitry to deliver programming and testing performance without overshoot or ground bounce for all device technologies. Pin drivers operate down to 1.8V so you'll be ready to program the full range of today's advanced low-voltage devices.

Modular design of JetProg allows adapting the programmer according to customer's needs either as very flexible universal programmer for laboratory or high efficient multiprogrammer in production line. Multiprogramming capability for most of supported devices is accomplished by using "multiple socket" modules.

Powerful pindrive provides logic level, pull-up/pull-down, clock, ground, one VCC supply and two programming supply and, certainly read, on each of all 48 pins independently. This advanced design give it the ability to program almost every programmable device in DIL up to 48 pins without adapter or family specific module. Support for today and tomorrow programmable devices gives engineers the freedom to choose the optimum device for new design.

JetProg isn't only programmer, but also **tester** of TTL/CMOS logic ICs and memories. Furthermore, it allows generate of user-definable **test pattern sequences**.

The programmer has on-board intelligence, comprise of powerful microcontroller system and support devices. JetProg has been designed for **multitasking operating systems** and is able to perform time-critical programming sequences independently of the PC operating system status and without being interrupted by any another parallel process running on the PC. Consequently, JetProg works without any problem on systems running Windows 3.x, Windows 95/98 and Windows NT/2000.

The programmer performs **device insertion test** (wrong or backward position) and contact check (poor contact pin-to-socket) before it programs each device. These capabilities, supported by overcurrent protection and signature-byte check help prevent chip damage due to operator error.

Built-in protection circuits eliminate damage of programmed device due to mains supply fluctuations, communication error or if PC is frozen. In event of such errors microcontroller in programmer performs, independently on the PC, exactly specified sequence of steps, so that programmed target device remains intact. Programmer's hardware offer enough resources for **selftest**, that control program is any time be able



to check pindrivers, present and correct level of all voltages, check the timing and communication between programmer and PC.

An optimally designed printed circuit minimizes negative programming effects at the socket (such as groundbouncing, supply voltage instability). All the inputs of the JetProg programmer, including the ZIF socket, connection to PC and power supply input, are **protected against ESD** to protect the programmer and programmed circuits against damage due ESD.

JetProg performs programming **verification** at the marginal level of supply voltage, which, obviously, improves programming yield, and guarantees long data retention.

The JetProg programmer is driven by an **easy-to-use control program** with pull-down menu, hot keys and online help. You can select target device by its class, by manufacturer or simply by typing a fragment of vendor name and/or part number. Besides the standard device-related commands (read, blank check, program, verify, erase) have been implemented some test functions (insertion test, signature-byte check), and some special functions (autoincrement, production mode - start immediately after insertion of chip into socket). The control program permits data manipulation within the buffer and between the buffer and files and performs automatic file format detection and conversion. The software is available in english and in more another languages.

The rich-featured **auto-increment function** of JetProg enables you to assign individual serial numbers to each programmed device. This function simply increments a serial number in the buffer each time a new device is inserted. Furthermore, the function enables the operator to read serial numbers and/or any programmed device identification signatures from a file.

Various **socket converters** are available for the JetProg, DIL to PLCC, DIL to PSOP, DIL to SOIC, DIL to TQFP and DIL to TSOP. Software controlled pin-swapping capability together with universal converters are what give the JetProg the ability to program the target device in any package type. For example, just one DIL44 to PLCC44 converter will allow the JetProg to handle any PLCC44 target device in the current device list.

Devices with more than 48 pins are supported by

- pindriver expansion module and universal single socket module
- simple special package converters



Note: *Simple adapter usually available from ELNEC or easily built by the user from information provided in the file "adapters.txt" included with the control program.*

It is important to remember that in most cases new devices require **only a software upgrade** since the JetProg has 48 true pin drivers, which can perform as required under program control. With our prompt service new devices can be added to the current list within hours!

Advanced design including protection circuits, original brand components and careful manufacturing allows us to provide a **three-year warranty** on parts and labor for the JetProg (limited 25,000-cycle warranty on ZIF socket).



JetProg elements

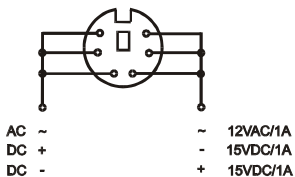
- ① DIL48 socket module with 48 pin ZIF socket
- ② LED indicator power/sleep
- ③ LED indicators for work result
- ④ YES! Button



- ⑤ Connector for PC <-> JetProg communication cable
- ⑥ Power supply connector
- ⑦ Internal use connector



Power supply connector



Note: When the JetProg programmer isn't active, control program switch the programmer into "sleep mode". In this state it consumes less than 1W of power. Therefore the programmer has no On/Off switch. When the power LED indicator glows with a low intensity the JetProg is in sleep mode.



Connecting JetProg to the PC

Switch off PC and programmer. Insert the communication cable included with your JetProg programmer package to a free printer port on your PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counter-connectors. This is very important. It may be uncomfortable to switch between printer cable and programmer cable, though it is not recommended to operate the JetProg programmer through a mechanical printer switch. Use of an electronic printer switch is impossible. But you can install a second multi-I/O in your computer, thus obtaining a supplementary printer port, say LPT2. So your printer may remain on LPT1 while the programmer on LPT2.

Switch on the PC.

Connect the mains connector of the power supply (or the wall-plug power supply itself) to a mains plug, then connect the mini-DIN connector to the programmer's connector labeled "12VAC". At this time all 'work result' LEDs (and 'POWER' LED) light up successive and then switch off. Once the POWER LED lights with low brightness then the JetProg programmer is ready to run.

Next run the control program for JetProg.



Caution! If you don't want to switch off your PC when connecting the JetProg, proceed as follows:

- **When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector**.
- **When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable**.

From JetProg's point of view the connecting and disconnecting sequence is irrelevant. Protection circuits on all programmer inputs keep it safe. **But think of your PC please.**

Note: Programmer's protection electronics protect the target device and the programmer itself against either short or long-term power failures and, partly, also against a PC failure. However, it is not possible to grant the integrity of the target device due to incorrect, user-selected programming parameters. Target device may be not destroyed by forced interruption of the control program (reset or switch-off PC), by removing the physical connection to the programmer, but the content of actually programmed cell may remains undefined. Don't unplug the target device from the ZIF socket during work with devices (LED BUSY shine).

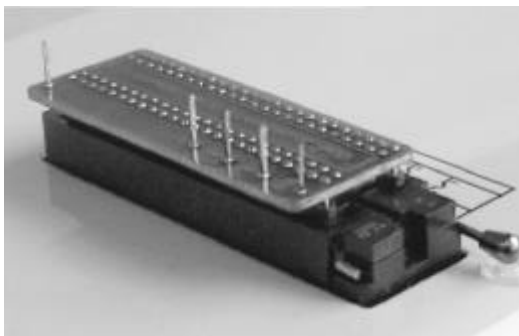
Problems related to the JetProg ⇔ PC interconnection, and their removing

If you have any problems with JetProg ⇔ PC interconnection, see section **Common notes** please.



Selftest and Calibration

If you feel that your programmer does not react according to your expectation, please run the programmer selftest using Diagnostic POD, enclosed with the standard delivery package. For optimal results with programmer we recommend you undertake every 6 months, an extended test and to check the calibration. See instructions for selftest in the TEST menu from PG4UW / PG4U.





Technical specification

HARDWARE

Base unit, DACs

- FPGA based IEEE 1284 slave printer port, up to 1MB/s transfer rate
- on-board powerful microprocessor (20MHz) supported by FPGA based state machine, 20MHz powered
- three D/A converters for VCCP, VPP1, and VPP2, controllable rise and fall time
- VCCP range 0..8V/1A
- VPP1, VPP2 range 0..26V/1A
- autocalibration
- selftest capability
- protection against surge and ESD on power supply input, parallel port connection
- banana jack for ESD wrist straps

Socket, pindriver

- pindrivers: 48 as standard, max. 256
- 1x VCC, 2x VPP can be connected to each pin
- perfect ground for each pin
- FPGA based TTL driver provides H, L, CLK, pull-up, pull-down on all pindriver pins
- analog pindriver output level selectable from 1.8 V up to 26V
- current limitation, overcurrent shutdown, power failure shutdown
- ESD protection on each pin of socket (IEC1000-4-2: 15kV air, 8kV contact)
- continuity test: each pin is tested before every programming operation

Socket, base configuration

- 48-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 48-pin

DEVICE SUPPORT

Programmer

- EPROM: NMOS/CMOS, 1702*, 2708*, 27xxx and 27Cxxx series, with 8/16 bit data width, full support for LV series
- EEPROM: NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx series, with 8/16 bit data width



- Flash EPROM: 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx series, from 256Kbit to 32Mbit, with 8/16 bit data width, full support for LV series
- serial E(E)PROM: 17Cxxx, 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx, NVM3060, MDAxxx series, full support for LV series
- PROM: Harris, National, Philips, Signetics, Tesla, TI
- NV RAM: Dallas DSxxx, SGS/Inmos MKxxx, SIMTEK STKxxx, XICOR 2xxx, ZMD U63x series
- PLD: PLD, CPLD series: Altera, Atmel, AMD-Vantis, Lattice, NS, Philips, TI, ...
- microcontrollers 48 series: 87x41, 87x42, 87x48, 87x49, 87x50 series
- microcontrollers 51 series: 87xx, 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx, all manufacturers, Philips 87C748.752 series
- microcontrollers Intel 196 series: 87C196 KB/KC/KD/KT/KR/...
- microcontrollers Atmel AVR: AT90Sxxxx, ATtiny series
- microcontrollers Microchip PICmicro: PIC12Cxxx, PIC16C5x, PIC16Cxxx, PIC17Cxxx, PIC18Cxxx series
- microcontrollers Motorola: 68HC11 series
- microcontrollers National: COP8xxx series
- microcontrollers NEC: uPD78Pxxx series
- microcontrollers SGS-Thomson: ST6xx series
- microcontrollers TI: MSP430 series
- microcontrollers ZILOG: Z86xxx series

Notes:

- *Devices marked * are obsolete, programming with additional module*
- *For all supported devices see actual **Device list***

I.C. Tester

- TTL type: 54,74 S/LS/ALS/H/HC/HCT series
- CMOS type: 4000, 4500 series
- static RAM: 6116.. 624000
- user definable test pattern generation

Package support

- package support includes DIP, PLCC, SDIP, SOIC, PSOP, TSOP and other
- support all devices in DIP with default socket
- support devices in PLCC28, PLCC32 and PLCC44 with universal adapters
- programmer is compatible with third-party adapters for non-DIP support



Programming speed

Note. These times strongly depend on PC speed, LPT port type and operating system free resources. Therefore are given values of two different PC configurations for comparison.

Device	Operation	Time A	Time B
27C010	programming and verify	24 sec	20 sec
AT29C040A	programming and verify	38 sec	28 sec
AM29F040	programming and verify	55 sec	49 sec
PIC16C67	programming and verify	18 sec	16 sec

Time A conditions: *Pentium MMX, 250 MHz, ECP/EPP, WIN98.*

Time B conditions: *Athlon, 750 MHz, ECP/EPP on PCI bus, WIN98.*

SOFTWARE

- Algorithms: only manufacturer approved or certified algorithms are used. Custom algorithms are available at additional cost.
- Algorithm updates: software updates are available approx. every 2 weeks, free of charge.
- Main features: revision history, session logging, on-line help, device and algorithm information

Device operations

- **standard:**
 - intelligent device selection by device type, manufacturer or typed fragment of part name
 - automatic ID-based selection of EPROM/Flash EPROM
 - blank check, read, verify
 - program
 - erase
 - configuration and security bit program
 - illegal bit test
 - checksum
- **security**
 - insertion test, reverse insertion check
 - contact check
 - ID byte check
- **special**
 - production mode (automatic start immediately after device insertion)
 - auto device serial number increment
 - statistic
 - count-down mode



Buffer operations

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print
- file load/save
- no download time because programmer is PC controlled
- automatic file type identification

Supported file formats

- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S-record, MOS, Exormax, Tektronix, ASCII-space-HEX
- Altera POF, JEDEC (ver. 3.0.A), eg. from ABEL, CUPL, PALASM, TANGO PLD, OrCAD PLD, PLD Designer ISDATA, etc.

PC system requirements

See section *Introduction/ PC requirements*

GENERAL

- operating voltage 12..15V AC/max. 1A (adapter included in package)
- power consumption max. 12W active, about 1.5W sleep
- dimensions 275L x 157W x 47H mm (10,8"L x 6,2"W x 1,85"H)
- weight (without external adapter) 1.5kg
- temperature 5°..40°C
- humidity 20%..80%, non condensing

Base configuration package includes

- JetProg, base unit
- JetProg, DIL48 socket module
- connection cable
- diagnostic POD for selftest
- anti-dust cover for ZIF socket
- power supply adapter 220VAC/12VAC/12W (120VAC/12VAC/12W for 120VAC mains)
- user manual
- software
- "PROGRAMMER PROBLEM REPORT" and "DEVICE PROBLEM REPORT" forms
- registration card
- transport case



Additional services

- Keep Current.
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.



LabProg+



Introduction

LabProg+ is a universal **programmer** that supports programmable integrated circuits or devices manufactured in various technologies. Powerful internal pin-driver electronics controls logic levels, pull-up/pull-down, clock, ground, one power supply and two programming supplies and is able to read all 48 pins independently. This advanced design gives LabProg+ the ability to handle almost every programmable device in DIL package up to 48 pins without any adapters and/or family modules. This design philosophy allows ELNEC to easily add new devices to the device list, giving you the freedom to implement the optimum device in your designs.

LabProg+ is also a **tester** of TTL/CMOS logic circuits and various memories. Furthermore, it can generate user-definable **test pattern sequences**. LabProg+ is a true universal and a true low-cost programmer, providing the best "value for money" in today's market.

LabProg+ works with almost any IBM compatible PC, AT or higher, portable or desktop personal computers. No special interface card is required to connect to the PC since LabProg+ uses the **standard parallel printer port**. The LabProg+ control program also supports bi-directional protocols for the parallel connection to the PC printer port providing fast and reliable **communication speed**.

The programmer has on-board intelligence and is controlled by powerful microcontroller system and support devices. LabProg+ has been designed for **multitasking operating systems** and is able to perform time-critical programming sequences independently of the PC operating system status and without being interrupted by any other parallel process running on the PC. Consequently, LabProg+ functions flawlessly on systems running Windows 3.x, Windows 95/98 and Windows NT/2000.

LabProg+ performs **device insertion test** (wrong or backward position) and contact check (poor contact pin-to-socket) before it programs any device. These capabilities, supported by current limit protection and signature-byte check, help prevent chip damage due to operator error.

Built-in protection circuits help prevent damage of the target device due to mains supply fluctuations, communication errors or if the PC operating system fails. In the event of such errors the LabProg+ performs independently of the PC exactly specified special sequences so that the target device remains intact. LabProg+ performs **selftest** (diagnostic tests), including



verification of pin-driver voltage/level, for accurate timing of the signals applied to the target device and for reliable communication with the PC.

LabProg+ incorporates optimal PCB design criteria to minimize unwanted effects at the pins of the target socket (such as ground-bouncing and supply/programming voltage glitches). All the inputs of the LabProg+, including the socket, are **protected against ESD** and whilst inserted the target device is also protected against ESD damage.

LabProg+ performs programming **verification** at the marginal level of supply voltage which, obviously, improves programming yield and guarantees long data retention.

The LabProg+ is driven by an **easy-to-use, control program** with pull-down menus, hot keys and on-line help. You can select the target device by its class, by manufacturer or simply by typing a fragment of vendor name or part number. Besides the standard device-related commands (blank check, read, program, verify) have been implemented some test functions (illegal bit test, insertion test), and some special functions (production mode - start the action immediately after insertion of device into ZIF socket). The control program permits data manipulation within the buffer and between the buffer and associated files, and performs automatic file format detection and conversion.

The control program provides an **auto-increment function** that enables you to assign individual serial numbers to each programmed device, this function simply increments a serial number in the buffer each time a new device is inserted in the socket. Furthermore, the function enables the operator to read serial numbers and/or any programmed device identification signatures from a file.

Various **socket converters** are available for the LabProg+, DIL to PLCC, DIL to PSOP, DIL to SOIC, DIL to TQFP and DIL to TSOP. Software controlled pin-swapping capability together with universal converters are what give the LabProg+ the ability to program the target device in any package type. For example, just one DIL44 to PLCC44 converter will allow the LabProg+ to handle any PLCC44 target device in the current device list. Most devices with more than 48 pins require a simple adapter usually available from ELNEC or easily built by the user from information provided in the file "adapters.txt" included with the control program.

It is important to remember that in most cases new devices require **only a software upgrade** since the LabProg+ has 48 true pin drivers, which can perform as required under program



control. With our prompt service new devices can be added to the current list within hours!

Advanced design, including protection circuits, original brand components and careful manufacturing allows us to provide a **three-year warranty** on parts and labor for the LabProg+ (limited 25,000-cycle warranty on ZIF socket).

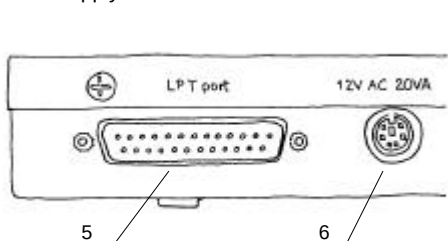


LabProg+ elements

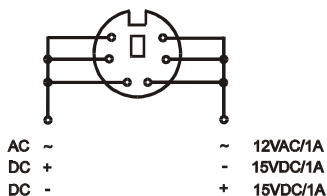
- ① 48 pin ZIF (Zero Insertion Force) socket
- ② LED indicator power/sleep
- ③ LED indicators for work result
- ④ YES! Button



- ⑤ Connector for PC <-> LabProg+ communication cable
- ⑥ Power supply connector



Power supply connector

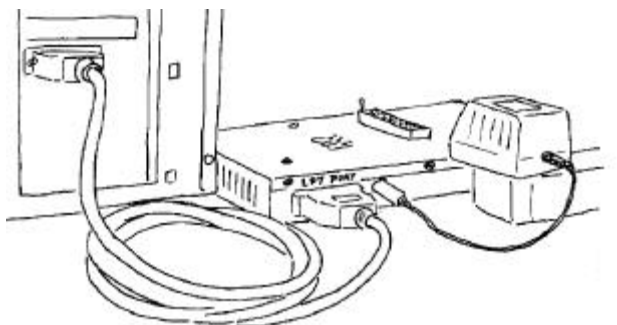


Note: When the LabProg+ programmer isn't active, control program switch the programmer into "sleep mode". In this state it consumes less than 1W of power. Therefore the programmer has no On/Off switch. When the power LED indicator glows with a low intensity the LabProg+ is in sleep mode.



Connecting LabProg+ to the PC

Switch off PC and programmer. Insert the communication cable included with your LabProg+ programmer package to a free printer port on your PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counter-connectors. This is very important. It may be uncomfortable to switch between printer cable and programmer cable, though it is not recommended to operate the LabProg+ programmer through a mechanical printer switch. Use of an electronic printer switch is impossible. But you can install a second multi-I/O in your computer, thus obtaining a supplementary printer port, say LPT2. So your printer may remain on LPT1 while the programmer on LPT2.



Switch on the PC.

Connect the mains connector of the power supply (or the wall-plug power supply itself) to a mains plug, then connect the mini-DIN connector to the programmer's connector labeled "12VAC". At this time all 'work result' LEDs (and 'POWER' LED) light up successive and then switch off. Once the POWER LED lights with low brightness then the LabProg+ programmer is ready to run.

Next run the control program for LabProg+.

Note: When the PC is switch off and you turn on programmer, LED maybe not blinking, before programmer maybe permanent on reset.



Caution! If you don't want to switch off your PC when connecting the LabProg+, proceed as follows:

- **When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector**.
- **When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable**.

From LabProg+'s point of view the connecting and disconnecting sequence is irrelevant. Protection circuits on all programmer inputs keep it safe. **But think of your PC please.**

Note: Programmer's protection electronics protect the target device and the programmer itself against either short or long-term power failures and, partly, also against a PC failure. However, it is not possible to grant the integrity of the target device due to incorrect, user-selected programming parameters. Target device may be not destroyed by forced interruption of the control program (reset or switch-off PC), by removing the physical connection to the programmer, but the content of actually programmed cell may remains undefined. Don't unplug the target device from the ZIF socket during work with devices (LED BUSY shine).

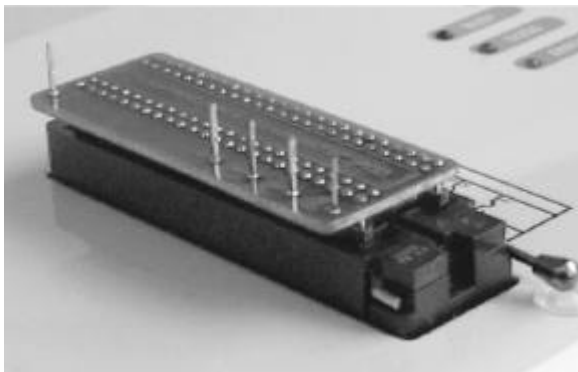
Problems related to the LabProg+ ↔ PC interconnection, and their removing

If you have any problems with LabProg+ ↔ PC interconnection, see section **Common notes** please.



Selftest and Calibration

If you feel that your programmer does not react according to your expectation, please run the programmer selftest using Diagnostic POD, enclosed with the standard delivery package. For optimal results with programmer we recommend you undertake every 6 months, an extended test and to check the calibration. See instructions for selftest in the TEST menu from PG4UW.





Technical specification

Socket, pin drivers and DACs

- 48-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 48-pin
- Three D/A converters for VCCP, VPP1, and VPP2, with controllable rise and fall time and current limitation
- TTL driver provides H, L, CLK, pull-up, pull-down, or tri-state on all 48 pins
- full support of Low Voltage circuits from 1.8 V up
- autocalibration

Device support

Programmer

- EPROM: NMOS/CMOS, 27xxx and 27Cxxx series, from 16Kbit to 32Mbit with 8/16 bit data width, full support for LV series
- EEPROM: NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx series, with 8 / 16 bit data width
- Flash EPROM: 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx series, from 256Kbit to 32Mbit, with 8 / 16 bit data width, full support for LV series
- serial E(E)PROM: 17Cxxx, 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx, NVM3060, MDAxxx series, full support for LV series
- PROM: Harris, NS, Philips, Tesla, TI
- NV RAM: Dallas DSxxx, SGS/Inmos, MKxxx series
- PLD: AMD PALCE, GALs, PEELs series
- microcontrollers 48 series: 87x41, 87x42, 87x48, 87x49, 87x50 series
- microcontrollers 51 series: 87xx, 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx, Philips 87C748..752 series
- microcontrollers Microchip PIC: PIC12xxx, PIC16xxx, PIC17Cxxx, PIC18xxx series
- microcontrollers Atmel AVR: AT90Sxxxx series
- microcontrollers NEC (uPD78Pxxx series), SGS-Thomson (ST6 series), Zilog (Z8 series), etc.

I.C. Tester

- TTL type: 54,74 S/LS/ALS/H/HC/HCT series
- CMOS type: 4000, 4500 series
- static RAM: 6116 .. 624000
- user definable test pattern generation



Package support

- package support includes DIP, PLCC, SDIP, SOIC, PSOP, TSOP, TQFP and other (ask for package converters)
- support all devices in DIP with default ZIF-48 socket
- support PLCC28, PLCC32 and PLCC44 with universal adapter (optional accessory, to be ordered separately)
- compatible with third-party adapters for non-DIP support

Programming speed

- M27C256, Program and verify, PC486/DX50: 10 seconds
- M27C010, Program and verify, PC586/100: 38 seconds
- AT29C040, Program and verify, PC586/100: 120 seconds

Device operations

- **standard:**
 - automatic ID-based selection of EPROM/Flash EPROM
 - blank check
 - read
 - program
 - verify
 - erase
 - configuration and security bit program
 - illegal bit test
 - checksum
- **security**
 - insertion test
 - contact check
 - ID byte check
- **special**
 - production mode (automatic start immediately after device insertion)
 - automatic device serial number incrementation

Buffer operations

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Supported file formats

- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-space-HEX
- JEDEC (ver. 3.0.A)



PC system requirements

See section *Introduction/ PC requirements*

General

- operating voltage 12..15V AC/max.1A (adapter 220VAC/12VAC included in package)
- power consumption - max. 12W in active mode, approx. 1W sleep mode
- dimensions 275x157x47 mm
- weight (without external adapter) 1.5 kg
- operating temperature 5°..40°C
- humidity 20%.80%, non condensing

Package included

- LabProg+ programmer
- connection cable
- diagnostic POD for selftest
- anti-dust cover to ZIF socket
- power supply adapter 220VAC/12VAC/12W (120VAC/12VAC for 120VAC mains)
- user manual
- 3.5" diskette with software
- "PROGRAMMER PROBLEM REPORT" and "DEVICE PROBLEM REPORT" forms
- registration card
- transport case

Additional services

- Keep Current
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.



PREPROM-02aLV



Introduction

PREPROM-02aLV is a small and powerful EPROM, EEPROM, Flash EPROM and serial EEPROM **programmer** and static RAM **tester**, designed for professional mobile applications. In addition, PREPROM-02aLV programmer with auxiliary modules support also microprocessors (MCS48, MCS51, PICmicro, AVR), GALs, etc. Generators for supply voltage and programming voltage are digitally controlled and level of H can be limited, therefore programmer can work with 'true LV' device too - from 2V. Performance, dimensions and speed of PREPROM-02aLV can be used both in maintenance and in production.

PREPROM-02aLV interfaces with the IBM PC, AT or higher, portable or desktop personal computers. Programmers allow you to directly connect to your PC through any standard parallel (printer) port - no special interface card is needed.

Built-in protection circuits eliminate damage of programmed device due to mains supply error, communication error or if PC is frozen. Programmer's hardware afford enough resources for selftest, that control program is any time be able to check pindrivers, present of all voltages, check the timing and communication between programmer and PC.

Programming verification is performed by the programmer at the marginal level of VCCP, which, obviously, improves programming yield, and guarantees long data retention.

The advantage of PREPROM-02aLV is its easy-to-use comfortable software with pull-down menu, hot keys and on-line help. You can select device by classes, by manufacturers or simply type device vendor's name and part number. The standard device operation functions (read, blank check, program, verify) have been completed with some test functions (ID check). The program facilitates the use of the buffer and files, including automatic file format detection and conversion.

The control program provides an **auto-increment function** that enables you to assign individual serial numbers to each programmed device, this function simply increments a serial number in the buffer each time a new device is inserted in the socket. Furthermore, the function enables the operator to read serial numbers and/or any programmed device identification signatures from a file.



The PREPROM-02aLV can make for use of DIL to PLCC, SOIC, PSOP socket converters.

Taking into consideration the PREPROM-02aLV programmer design, protective circuits, and the components used, the manufacturer is able to provide a three-year warranty on parts and labor for the programmer (limited 25,000-cycle warranty on the ZIF socket).



PREPROM-02aLV diagram

- ① 32 pin ZIF socket
- ② LED power/sleep
- ③ LED, which indicate work result
- ④ YES! button



- ⑤ communication connector, for PC ⇔ PREPROM-02aLV cable connection
- ⑥ connector for power supply connection

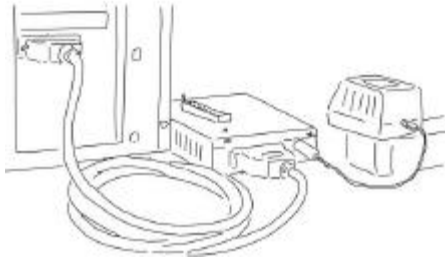


Note: Due to low power consumption of PREPROM-02aLV in inactive state, it doesn't require power switch.



Connecting PREPROM-02aLV programmer to PC

Switch off the PC and programmer. Insert the connection cable, included in the PREPROM-02aLV programmer delivery, to the free printer port of PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counterconnectors. This is very important mainly for the connector to programmer. Though replacing the printer cable by the programmer cable is uncomfortable, it is not recommended to operate the PREPROM-02aLV programmer through a mechanical printer switch. Use of an electronic printer switch isn't possible.



Connect the mains connector of the power supply (or wall-plug power supply self) to a mains plug, connect the connector to the programmer's connector labeled 12VDC. Then, on the programmer lights up LED POWER and the programmer PREPROM-02aLV is ready to run. Next switch on the PC and run the control program.

Caution! *If you don't want to switch off your PC when connecting the PREPROM-02aLV, proceed as follows:*

- ***When connecting*** the programmer to the PC: ***FIRST*** insert the ***communications cable*** and ***THEN*** the ***power-supply connector***.
- ***When disconnecting*** the programmer from the PC: ***FIRST*** disconnect the ***power-supply connector*** and ***THEN*** the ***communication cable***.



Problems related to the PREPROM-02aLV ⇔ PC interconnection, and their removing

If you have any problems with PREPROM-02aLV ⇔ PC interconnection, see section **Common notes** please.

Note for laptop and PREPROM-02aLV users

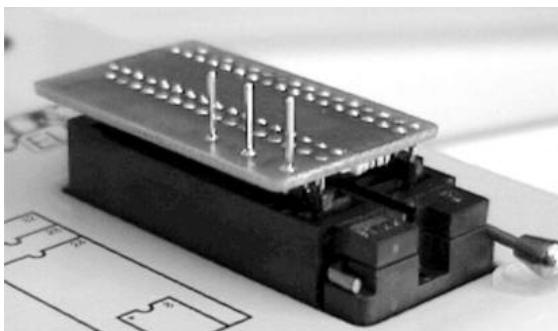
Since the PREPROM-02aLV programmer is supplied with 12V, notebook and/or laptop users may operate their PREPROMs-02aLV everywhere this voltage is available, e.g. in a car. Ten series-connected NiCd batteries can supply the programmer for more than one hour at full performance. Unwanted swapping has no effects - the programmer is protected against the reversal of the power supply poles. The supply cable you can make yourself, polarity is:





Selftest and calibration

If you have feeling, that your programmer doesn't behave according to yours expectations (but at least every 3 month), please run PREPROM-02aLV selftest using Diagnostic POD, that contains standard programmer delivery. For optimal fruit of work with PREPROM-02aLV programmer is recommended every 6 months (no more) to make extended test and check the calibration. All this activities you can find in TEST menu.





PREPROM-02aLV specification

Socket, pin drivers and DACs

- 32-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 32-pin
- two D/A converters for VCCP and VPP, with controllable rise/fall time and current limitation
- TTL driver provides H, L and read all 32 pins
- full support of Low Voltage circuits from 2.0 V

Device support

Programmer

- EPROM: NMOS/CMOS, 27xxx and 27Cxxx series, with 8/16* bit data width
- EEPROM: NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx series, with 8/16* bit data width
- Flash EPROM: 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx series, from 256Kbit to 32Mbit, with 8/16* bit data width
- serial E(E)PROM: 17Cxxx*, 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx, NVM3060, MDAXxx* series
- NV RAM: Dallas DSxxx, SGS/Inmos MKxxx series
- PLD*: AMD PALCE, GALs, PEELs series
- microcontrollers 48 series*: 87x41, 87x42, 87x48, 87x49, 87x50 series
- microcontrollers 51 series*: 87xx, 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx, Philips 87C748..752 series
- microcontrollers Microchip PIC*: PIC12Cxxx, PIC16C5x, PIC16Cxxx, PIC17Cxxx series
- microcontrollers Atmel AVR*: AT90Sxxxx series
- microcontrollers NEC*: uPD78Pxxx series
- note: * - programming with additional module

I.C. Tester

- static RAM: 6116 .. 624000

Programming speed

- M27C256, programming + verifying, PC486/DX50 : 16 sec
- M27C010, programming + verifying, PC586/100 : 40 sec



Device operations

- **standard:**
 - automatic ID-based selection of EPROM/Flash EPROM
 - blank check
 - read
 - program
 - verify
 - erase
 - configuration and security bit program
 - illegal bit test
- **security:**
 - contact check
 - ID byte check
- **special**
 - auto device serial number increment

Buffer operations

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Supported file formats

- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-space-HEX
- JEDEC

PC system requirements

See section *Introduction/ PC requirements*

General

- operating voltage 12VDC/max. 500mA (adapter 220VAC/12VDC included in package)
- power consumption 6W max.
- dimensions 160x110x50 [mm] (6.3x4.3x2.0 [inch])
- weight (without external adapter) ca. 650g
- temperature 5° ÷ 40°C
- humidity 20%..80%, non condensing



Package included

- PREPROM-02aLV programmer
- 1.8m (6 ft.) connecting cable PREPROM-02aLV to (IBM) PC
- wall plug adapter 220V/50Hz to 12V DC
- diagnostic POD, that enable programmer's selftest
- diskette with control program
- user manual
- forms "PROGRAMMER PROBLEM REPORT" and "DEVICE PROBLEM REPORT"
- registration card
- transport packing

Additional services

- Keep Current
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.



51&AVRprog



Introduction

51&AVRprog is little and powerful portable programmer for MCS51 series and Atmel AVR microcontrollers. 51&AVRprog enables also programming serial EEPROM with interface types IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx). The programmer is equipped by DIL ZIF 40pin socket. The quality of programmer is completed by comfortable control program.

51&AVRprog interfaces with the IBM PC, AT or higher, portable or desktop personal computers. A programmer allows you to directly connect to your PC through any standard parallel (printer) port - no special interface card is needed.

The advantage of 51&AVRprog is its easy-to-use comfortable software with pull-down menu, hot keys and on-line help. You can select device by classes, by manufacturers or simply type device vendor's name and part number. The standard device operation functions (read, blank check, program, verify) have been completed with some test functions. The program facilitates the use of the buffer and files, including automatic file format detection and conversion.

The 51&AVRprog can make for use of DIL to PLCC and SOIC socket converters.

Warning: *Jumper must be in position JP2-JP3 for MCS51 devices and JP3-JP4 for AVR family device. Jumper is accessible after removing bottom part of programmer case. Default jumper position is JP2-JP3 (for MCS51 family).*



Connecting 51&AVRprog programmer to PC

Switch off the PC and programmer. Insert the connection cable, included in the 51&AVRprog programmer delivery, to the free printer port of PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counterconnectors. This is very important mainly for the connector to programmer. Though replacing the printer cable by the programmer cable is uncomfortable, it is not recommended to operate the 51&AVRprog programmer through a mechanical printer switch. Use of an electronic printer switch isn't possible.

Connect the mains connector of the power supply (or wall-plug power supply self) to a mains plug, connect the connector to the programmer's connector labeled 12VDC. Then, on the programmer lights up LED POWER and the programmer 51&AVRprog is ready to run. Next switch on the PC and run the control program.

Caution! *If you don't want to switch off your PC when connecting the 51&AVRprog , proceed as follows:*

- **When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector**.
- **When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable**.

Problems related to the 51&AVRprog ⇔ PC interconnection, and their removing

If you have any problems with 51&AVRprog ⇔ PC interconnection, see section **Common notes** please.

Manipulation with the programmed device

After selection of desired device for your work, you can insert into the open ZIF socket (the lever is up) and close socket (the lever is down). The correct orientation of the programmed device in ZIF socket is shown on the picture near ZIF socket on the programmer's cover. The programmed device is necessary to insert into the socket also to remove from the socket when LED BUSY light off.

Warning! *51&AVRprog programmer hasn't protection devices, which protect the content of programmed device against critical situations, for example power failures and PC failure*



(interrupted cable...). Moreover, an device is usually destroyed in the programming mode due to forced interruption of the control program run (Reset or switching the computer off) due to removing the connecting cable, or unplugging the programmed device from the ZIF socket. Incorrectly placed device in the ZIF socket can cause its damage or destruction.



51&AVRprog specification

Socket, pin drivers and DACs

- 40-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 40-pin
- TTL driver provides H, L and read for all pins

Device support

- microcontrollers 51 series: 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx series from Atmel, Intel, Philips, ... 20/40 pins
- microcontrollers Atmel AVR: AT90Sxxxx series (parallel mode), 8/20/40 pins
- serial E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx series

Device operations

- **standard:**
 - blank check
 - read
 - program
 - verify
 - erase
 - configuration and security bit program
- **security:**
 - ID byte check
- **special:**
 - auto device serial number increment

Buffer operations

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Supported file formats

- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-space-HEX, Tektronix



PC system requirements

See section *Introduction/ PC requirements*

General

- operating voltage 15÷20V DC/200mA
- power consumption 3W max.
- dimensions 132x66x30 [mm] (5.2x2.6x1.2 [inch])
- weight (without external adapter) ca. 200g
- temperature 5° ÷ 40°C
- humidity 20%..80%, non condensing

Package included

- 51&AVRprog programmer
- 1.3m (4.3 ft.) connecting cable 51&AVRprog to (IBM) PC
- wall plug adapter 12V DC/ 500mA, unstabilized (output 15V if load 200mA)
- diskette with control program
- user manual
- forms "PROGRAMMER PROBLEM REPORT" and "DEVICE PROBLEM REPORT"
- registration card
- transport packing

Additional services

- Keep Current
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.



PIKprog



Introduction

PIKprog is little and powerful portable programmer for Microchip PIC series of microcontrollers. PIKprog enables also programming serial EEPROM with interface types IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx). The programmer is equipped by DIL ZIF 40pin socket. The quality of programmer is completed by comfortable control program.

PIKprog interfaces with the IBM PC, AT or higher, portable or desktop personal computers. Programmer allows you to directly connect to your PC through any standard parallel (printer) port - no special interface card is needed.

The advantage of PIKprog is its easy-to-use comfortable software with pull-down menu, hot keys and on-line help. You can select device by classes, by manufacturers or simply type device vendor's name and part number. The standard device operation functions (read, blank check, program, verify) have been completed with some test functions. The program facilitates the use of the buffer and files, including automatic file format detection and conversion.

For PIKprog programmer are available DIL to SOIC socket converters.



Connecting PIKprog programmer to PC

Switch off the PC and programmer. Insert the connection cable, included in the PIKprog programmer delivery, to the free printer port of PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counterconnectors. This is very important mainly for the connector to programmer. Though replacing the printer cable by the programmer cable is uncomfortable, it is not recommended to operate the PIKprog programmer through a mechanical printer switch. Use of an electronic printer switch isn't possible.

Connect the mains connector of the power supply (or wall-plug power supply self) to a mains plug, connect the connector to the programmer's connector. Then, on the programmer lights up LED POWER and the programmer PIKprog is ready to run. Next switch on the PC and run the control program.

Caution! *If you don't want to switch off your PC when connecting the PIKprog, proceed as follows:*

- **When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector**.
- **When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable**.

Problems related to the PIKprog ⇔ PC interconnection, and their removing

If you have any problems with PIKprog ⇔ PC interconnection, see section **Common notes** please.

Manipulation with the programmed device

After selection of desired device for your work, you can insert into the open ZIF socket (the lever is up) and close socket (the lever is down). The correct orientation of the programmed device in ZIF socket is shown on the picture near ZIF socket on the programmer's cover. The programmed device is necessary to insert into the socket also to remove from the socket when LED BUSY light off.

Warning! *PIKprog programmer hasn't protection devices, which protect the content of programmed device against critical situations, for example power failures and PC failure (interrupted cable...). Moreover, an device is usually destroyed*



in the programming mode due to forced interruption of the control program run (Reset or switching the computer off) due to removing the connecting cable, or unplugging the programmed device from the ZIF socket. Incorrectly placed device in the ZIF socket can cause its damage or destruction.



PIKprog specification

Socket, pin drivers and DACs

- 40-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 40-pin
- TTL driver provides H, L and read for all pins

Device support

- microcontrollers Microchip PIC series: 12xxx, 14xxx, 16xxx, 8/16/18/28/40 pins
- Serial E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx series

Device operations

- **standard:**
 - blank check
 - read
 - program
 - verify
 - erase
 - configuration and security bit program
- **security:**
 - ID byte check
- **special:**
 - auto device serial number increment

Buffer operations

- view/edit, find/replace
- fill/copy, move, byte swap, word/dword split
- checksum (byte, word)
- print

Supported file formats

- unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-space-HEX, Tektronix



PC system requirements

See section *Introduction/ PC requirements*

General

- operating voltage 15÷20V DC/max. 200mA
- power consumption 3W max.
- dimensions 132x66x30 [mm] (5.2x2.6x1.2 [inch])
- weight (without external adapter) ca. 200g
- temperature 5° ÷ 40°C
- humidity 20%..80%, non condensing

Package included

- PIKprog programmer
- 1.3m (4.3 ft.) connecting cable PIKprog to (IBM) PC
- wall plug adapter 12V DC/ 500mA, unstabilized (output 15V if load 200mA)
- diskette with control program
- user manual
- forms "PROGRAMMER PROBLEM REPORT" and "DEVICE PROBLEM REPORT"
- registration card
- transport packing

Additional services

- Keep Current
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.



SEEplog



Introduction

SEEplog is universal programmer of all serial EEPROM in 8-pin DIL package. SEEplog programs EEPROM with interface IIC, SPI and Microwire, and also specialty as for example digital thermometers. The programmer supports LV (3.3V) devices too. The interest is using of programmer as the IIC bus master - by connecting directly to device via in-circuit programming cable (optional accessories). This way enabled reading and programming of serial EEPROM directly in circuit.

SEEplog interfaces with the IBM PC, AT or higher, portable or desktop personal computers. Programmer allows you to directly connect to your PC through any standard parallel (printer) port - no special interface card is needed.

The advantage of SEEplog is its easy-to-use comfortable software with pull-down menu, hot keys and on-line help. You can select device by classes, by manufacturers or simply type device vendor's name and part number. The standard device operation functions (read, blank check, program, verify) have been completed with some test functions. The program facilitates the use of the buffer and files, including automatic file format detection and conversion.

For SEEplog programmer are available DIL to SOIC socket converters.



Connecting SEEprog programmer to PC

Switch off the PC and programmer. Insert the connection cable, included in the SEEprog programmer delivery, to the free printer port of PC. If your computer is equipped with only one printer port, substitute the programmer cable for the printer cable. Connect the opposite cable end to the programmer. Screw on both connectors to counterconnectors. This is very important mainly for the connector to programmer. Though replacing the printer cable by the programmer cable is uncomfortable, it is not recommended to operate the SEEprog programmer through a mechanical printer switch. Use of an electronic printer switch isn't possible.

Connect the mains connector of the power supply (or wall-plug power supply self) to a mains plug, connect the connector to the programmer's connector labeled 12VDC. Then, on the programmer lights up LED POWER and the programmer SEEprog is ready to run. Next switch on the PC and run the control program.

Caution! *If you don't want to switch off your PC when connecting the SEEprog, proceed as follows:*

- **When connecting** the programmer to the PC: **FIRST** insert the **communications cable** and **THEN** the **power-supply connector**.
- **When disconnecting** the programmer from the PC: **FIRST** disconnect the **power-supply connector** and **THEN** the **communication cable**.

Problems related to the SEEprog ↔ PC interconnection, and their removing

If you have any problems with SEEprog ↔ PC interconnection, see section **Common notes** please.

Manipulation with the programmed device

After selection of desired device for your work, you can insert into the open ZIF socket (the lever is up) and close socket (the lever is down). The correct orientation of the programmed device in ZIF socket is shown on the picture near ZIF socket on the programmer's. The programmed device is necessary to insert into the socket also to remove from the socket when LED BUSY light off.

Warning! *SEEprog programmer hasn't protection devices, which protect the content of programmed device against critical*



situations, for example power failures and PC failure (interrupted cable...). Moreover, an device is usually destroyed in the programming mode due to forced interruption of the control program run (Reset or switching the computer off) due to removing the connecting cable, or unplugging the programmed device from the ZIF socket. Incorrectly placed device in the ZIF socket can cause its damage or destruction.



SEEprog specifications

Socket and control of pins:

- DIL/ZIF socket (300mil)
- each pin is possible to set in position Low and Pull-up
- from each pin is possible to read
- support of Low Voltage devices

For work with serial EEPROM are used nearest 8 pins to lever, unused pins are not connected.

Supported devices:

- EEPROM IIC (24Cxxx)
- EEPROM Microwire (93Cxxx)
- EEPROM SPI (25Cxxx)
- specially devices (digital thermometers, ...)

Device operations:

- **standard:**
 - blank check
 - read
 - program
 - verify
 - checksum
- **security:**
 - device insertion/contact check ZIF socket - device

File format conversion

- binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, ASCII - space - HEX

PC systems requirements

See section *Introduction/ PC requirements*

General:

- power supply: 8V...20V/DC, max.50mA
- dimensions: 132x66x30 [mm]
- mass: cca 150g
- temperature range: 5..40°C



Package included

- SEEprog programmer
- 1.3m (4.3 ft.) connecting cable SEEprog to (IBM) PC
- wall plug adapter 12V DC/ 500mA, unstabilized (output 15V if load 200mA)
- diskette with control program
- user manual
- forms "PROGRAMMER PROBLEM REPORT" and "DEVICE PROBLEM REPORT"
- registration card
- transport packing

Additional services

- Keep Current
- AlgOR
- free technical support (phone/fax/e-mail).
- free lifetime software update via Web site.



Software



The programmer software

The programmer package contains a floppy disk(s) or CD with the control program, useful utilities and additional information. We recommend making a copy of the original disk(s). The permission to freely copy the content of the floppy is granted in order to demonstrate how ELNEC's programmers works. Differences and modifications to this manual (if they exist) may be found in README_P.TXT file on the included floppy.

Installing of programmer software

Installing the programmer software is very easy. Start the Install program from the floppy disk, which will guide you through the installation process and which will do all the necessary steps before you can first run the control program.

If you use the operating system **DOS**, run **Install** program from diskette **Setup/Install (Disk 1/2)**. If you use the operating system **MS WINDOWS**, run **Setup** program from diskette **Setup/Install (Disk 1/2)**.

Programs PG4U.EXE (for DOS) and PG4UW.EXE (for Windows) are common control programs for all ELNEC's programmers. We guarantee running of these programs under all of above mentioned operating systems without any problems. Also background operation under Windows is error-free.

For DOS versions of the control program running under WINDOWS 3.11/95/98, we suggest the following: (adjust "Properties" on the icon on your screen):

- Run the control program in **full-screen DOS** (in text mode). In graphic mode the control program might be somewhat slower.
- If you want the control program to run in the **background**, it is necessary to set: (WIN95/98) RIGHT mouse button on the DOS icon/ Properties/ Others/ Background/ Enable; (WIN3xx) in the appropriate PIF file, set the option to Background

New versions of programmer software

In order to exploit all the capabilities of programmer we recommend using the latest version of PG4U/PG4UW (see appendix C - Keep-Current Service). You may download the latest version of programmer software (file PG4UARC.EXE /PG4UWARC.EXE) from our Internet site **www.elnec.com**, part download. You may also obtain a floppy disk or CD with this file by snail-mail (a mailing charge will apply).



Upgrading the programmer software

Copy PG4UARC.EXE / PG4UWARC.EXE to a temporary directory then launch it. After extraction you will see all available files needed for the installation process. Then redo a standard installation (run the Setup / Install program). You may delete all files from the temporary folder after the installation process is complete.

Using the programmer software

The control program delivered by ELNEC, included on the floppy or CD in your package, is granted to be free from any viruses at the moment of delivery. To increase their safety our programs include a special algorithm for detecting possible virus infections. Therefore keep the delivered floppy write protected.

Run the control program

From command line of MS/PC-DOS write:

PG4U

in Windows environment: double click to icon PG4UW.

After start, control program PG4U / PG4UW automatically scan all existing ports and search for the connected any ELNEC's programmer. Program PG4U/PG4UW is common for all the ELNEC's programmers, hence program try to find all supported (JetProg, LabProg+, PREPROM-02aLV, 51&AVRprog, PIKprog and SEEprog) programmers.

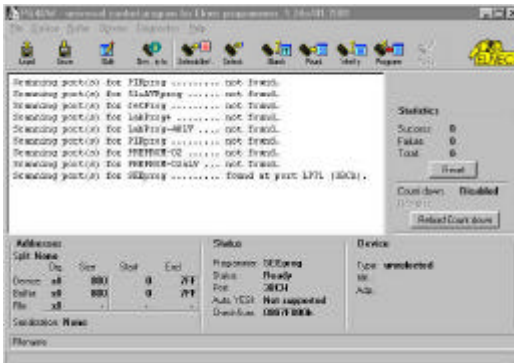
Notes: *When the PG4U/PG4UW program is started, program is checked for its integrity. Than the program display a standard user menu and waits for your instructions.*

If the control program cannot communicate with the programmer, an error message appears on the screen, including error code and description of possible reasons (disconnected programmer, bad connection, power supply failure, incompatible printer port, ...). Eliminate the error source and press any key. If error condition still exist, the program resumes its operation in the demo mode and access to the programmer is not possible. If you cannot find the cause of the error, follow the instructions in **Troubleshooting** section. In addition, the control program checks communication with programmer prior to any operation with the programmed device.

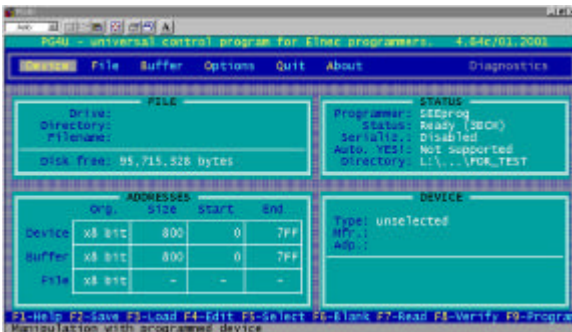


Description of the user screen

Windows program PG4UW



DOS program PG4U



Header bar

the name, copyright statement and version of the PG4U/PG4UW the control program

Menu bar

list of basic functions

File window/File name

information on the currently loaded file in buffer

Status window

information about the status of the programmer and PG4U/PG4UW

Addresses window

organization, size, start and end addresses of the target device, buffer and file

Device window

all relevant information about the current target device

Help bar

a brief description of selected command



Menu selection is carried out in the normal GUI fashion - either by cursor moving plus pressing **<Enter>**, or by typing the highlighted letter in the wanted menu or - of course - by mouse. Hot-keys are available for even quicker selection of intensely used commands.

Note: *Data entered through keyboard is in HEX format, excepting ASCII blocks in Buffer/View/Edit command.*

List of hot keys

<F1>	Help	Calls Help
<F2>	Save	Save file
<F3>	Load	Load a file into the buffer
<F4>	Edit	Viewing/editing of buffer
<F5>	Select/default	Target-device selection from 10 last selected devices list
<Alt+F5>	Select/manual	Target-device selection by typing device/vendor name
<Ctrl+F5>	Select/type	Target-device selection by device family
<Shift+F5>	Select/vendor	Target-device selection by device manufacturer
<F6>	Blank	Blank check
<F7>	Read	Reads device's content into the buffer
<F8>	Verify	Compares contents of the target device with the buffer
<F9>	Program	Programs target device
<Alt+Q>	Exit without save	Terminates the PG4U / PG4UW
<Alt+X>	Exit and save	Terminates the PG4U / PG4UW and saving settings too
<Ctrl+F1>		Displays additional information about current device
<Ctrl+F2>	Erase	Fill's the buffer with a given value
<Ctrl+Shift+F2>		Fill's the buffer with random values.



File

This submenu is used for source files manipulation, settings and viewing directory, changes drives, changes start and finish address of buffer for loading and saving files by **binary**, **MOTOROLA**, **MOS Technology**, **Intel (extended) HEX**, **Tektronix**, **ASCII space**, **JEDEC**, and **POF** format.

File / Load

Analyse file format and loads the data from specified file to the buffer. You can choose the format desired (**binary**, **MOTOROLA**, **MOS Technology**, **Tektronix**, **Intel (extended) HEX**, **ASCII space**, **JEDEC** and **POF**). The control program stores a last valid mask for file listing. You can save the mask into the config. file by command **Options / Save options**.

In the group box "File format recognition" in "Load file" dialog, user can select automatic or manual recognition of file format which has to be loaded.

The reserved key **<F3>** will bring out this menu from any menu and any time.

File / Save

Saves data in the buffer, which has been created, modified, or read from a device onto a specified disk. You can choose the format desired (**binary**, **MOTOROLA**, **MOS Technology**, **Tektronix**, **Intel (extended) HEX**, **ASCII space**, **JEDEC** and **POF**).

The reserved key **<F2>** will bring out this menu from any menu and any time.

File / Load encryption table

This command loads the data from binary file from disk and it saves them into the part of memory, reserved for an encryption (security) table.

File / Save encryption table

This command writes the content of the memory's part, reserved for an encryption table, into the file on the disk as a binary data.

File / Exit without save

Terminate program and exit to the operation system - without save settings.



File / Exit and save

Terminate program and exit to the operation system - save settings to INI file.



Device

The functions for a work with selected programmable devices - device select, read data from device, device blank check, device program, device verify and device erase.

Device / Select from default devices

This window allows selecting the desired type of the device from list of default devices. This one is a cyclic buffer in which are stored last 10 selected devices including its device options. This list is saved to disk by command **File / Exit and save**.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules), that supported this device. You can find here a package information and other general information about current device too.

Use a **<Del>** key for delete of current device from list of default devices. There isn't possible to empty this list, if you repeat this access. The last device stays in buffer and the **<Del>** key isn't accepted.

Device / Select device ...

This window allows selecting the desired type of the device from all devices supported by current programmer. It is possible to choose device by **name**, by **type** or by **manufacturer**.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device / Select from default devices** command.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules), that supported this device. You can find here a package information and other general information about current device too.

Select device ... / All

This window allows selecting the desired type of the device from all devices supported by current programmer. Supported devices are displayed in a list box.

Device can be select by double click on a line from list with desired manufacturer name and device number or by entering manufacturer name and/or device number in a search box (use



a key **<Space>** as a separation character) and press **<Enter>** or click **OK** button.

Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device / Select from default devices** command.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) which supported this device. You can find here a package information and other general information about current device too.

Select device ... / Only selected type

This window allows selecting the desired type of the device. At the first - you must select a device type (e.g. EPROM) and device subtype (e.g. 64Kx8 (27512)), using mouse or cursor keys. It will cause a list of manufacturers and devices will be displayed.

Device can be select by double click on a line from list with desired manufacturer name and device number or by entering manufacturer name and/or device number in a search box (use a key **<Space>** as a separation character) and press **<Enter>** or click **OK** button.

Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device / Select from default devices** command.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) which supported this device. You can find here a package information and other general information about current device too.

Select device ... / Only selected manufacturer

This window allows selecting the desired device type by manufacturer. First select a required manufacturer in Manufacturer box using mouse or cursor keys. It will cause a list of selected manufacturer devices will be displayed.

Device can be select by double click on a line from list with desired manufacturer name and device number or by entering



device number in a search box (use a key **<Space>** as a separation character) and press **<Enter>** or click **OK** button.

Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device / Select from default devices** command.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) which supported this device. You can find here a package information and other general information about current device too.

Device / Select EPROM by ID

Use this command for autoselect an EPROM as active device by reading the device ID. The programmer can automatically identify certain EPROM's by the reading the manufacturer and the device-ID that are burnt into the chip. This only applies to EPROM that supports this feature. If the device does not support a chip ID and manufacturer's ID, a message will be displayed indicating this as an unknown or not supported device.

If more EPROMs with identical chip ID and manufacturer's ID were detected, the list of these devices will be displayed. A corresponding device can be choosed from this list by selecting its number (or manufacturer name) from list and press **<Enter>** (or click **OK** button). Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

WARNING: *This time the control program support only EPROM's with 28 and 32 pins. Any of programmers determines pins number automatically. For other programmers you must enter this number manually.*

The programmer applies a high voltage to the appropriate pins on the socket. This is necessary to enable the system to read the device ID. Do not insert into the socket a device that is not an EPROM. It may be damaged when the programmer applies the high voltage.

We don't recommend apply this command to 2764 and 27128 EPROM types, because most of them ID not supports.



Device / Device options

All settings of this menu are used for programming process, serialization and associated file control.

Device / Device options / Operation options

All settings of this command are used for programming process control. This is a flexible environment which content items associated with current device and programmer type. Items, which are valid for the current device but aren't supported by current programmer, are disabled. These settings are saving to disk along with associated device by **File / Exit and save** command.

List of items:

- group Addresses:

device start address (default 0)
device end address (default device size-1)
buffer start address (default 0)

- group Insertion test:

insertion test (default ENABLE)
check ID bytes (default ENABLE)

- group Command execution:

blank check before programming (default DISABLE)
erase before programming (default DISABLE)
verify after reading (default ENABLE)
verify after programming (ONCE, TWICE)
verify options (nominal VCC 5%, nominal VCC 10%,
VCCmin VCCmax)

Device / Device options / Serialization

Serialization is special mode of program. When a serialization mode is activated, a specified value is automatically inserted on predefined address into buffer before programming each device. When more devices are programmed one by one, the serial number value is changed for each device automatically and inserted into buffer before programming device, so each device has unique serial number.

There are two types of serialization:

- Incremental mode
- From file mode

If a new device is selected, the serialization function is set to a default state i.e. disabled.



Actual serialization settings for actually selected device are saving to disk along with associated device by **File / Exit and save** command.

When incremental mode is active following actual settings are saved to configuration file: address, size, serial value, incremental step and settings of modes ASCII / BIN, DEC / HEX, LS byte / MS Byte first.

When from-file mode is active following actual settings are saved to configuration file: name of input serialization file and actual label, which indicates the line with actual serial number in input file.

Device / Device options / Serialization / Incremental mode

The Incremental mode enables to assign individual serial numbers to each programmed device. A starting number entered by user will be incremented by specified step for each device program operation and loaded in selected format to specified buffer address prior to programming of each device.

There are following options, that user can modify for incremental mode:

S / N size

S / N size option defines the number of bytes of serial value which will be written to buffer. For Bin (binary) serialization modes values 1-4 are valid for S / N size and for ASCII serialization modes values 1-8 are valid for S / N size.

Address

Address option specifies the buffer address, where serial value has to be written. Note that address range must be inside the device start and device end addresses. Address must be correctly specified so the last (highest or lowest) byte of serial value must be inside device start and device end address range.

Start value

Start value option specifies the initial value, from which serialization will start. Generally the max. value for serialization is \$1FFFFFFF in 32 bit long word.

When the actual serial value exceeds maximum value, three most significant bits of serial number are set to zero. After this action the number is always inside 0..\$1FFFFFFF interval (this is basic style of overflow handling).

Step

Step options specifies the increment step of serial value incrementation.

**S / N mode**

S / N mode option defines the form in which serial value has to be written to buffer. Two options are available:

- ASCII
- Bin

ASCII - means the serial number is written to buffer as ASCII string. For example number \$0528CD is in ASCII mode written to buffer as 30h 35h 32h 38h 43h 44h ('0' '5' '2' '8' 'C' 'D'), i.e. six bytes.

Bin - means the serial number is written directly to buffer. If the serial number has more than one byte length, it can be written in one of two possible byte orders. The byte order can be changed in „Save to buffer“ item.

Style

Style option defines serial number base. There are two options:

- Decimal
- Hexadecimal.

DECIMAL numbers are entered and displayed using the characters '0' through '9'.

HEXADECIMAL numbers also use characters 'A' through 'F'.

The special case is Binary Dec, that means BCD number style. BCD means the decimal number is stored in hexadecimal number, i.e. each nibble must have value from 0 to 9. Values A to F are not allowed as nibbles of BCD numbers.

Select the base in „Style“ options before entering numbers of serial start value and step.

Save to buffer

Save to buffer option specifies the serial value byte order to write to buffer. This option is used for Bin S / N mode (for ASCII mode it has no effect).

Two options are available:

- LSByte first (used by Intel processors) will place the Least Significant Byte of serial number to the lowest address in buffer.
- MSByte first (used by Motorola processors) will place the Most significant Byte first to the lowest address in buffer.

Device / Device options / Serialization / From file mode

Using the From-file method, serial values are read from the user specified input file and written to buffer on address specified in input file.

There are two user options: File name and Start label.

**File name**

File name option specifies the file name from which serial addresses and values will be read. The input file for From file serialization must have special format, which is described in From file serialization file format below.

Start label

Start label defines the start label in input file. The reading of serial values from file starts from defined start label.

From file serialization file format

From file serialization input file includes addresses and arrays of bytes defining buffer addresses and data to write to buffer. Input file has text type format, which structure is:

```
[label1] addr byte0 byte1 .. byten
...
[labeln] addr byte0 byte1 .. bytem , addr byte0 byte1 ... bytek
\          /          \          /
|          |          |          |
basic part optional part

; Comment
```

meaning is:

basic part

Basic part defines buffer address and array of bytes to write to buffer. Basic part must be always defined after label in line.

optional part

Optional part defines the second array of bytes and buffer address to write to buffer. One optional part can be defined after basic part of data.

label1, labeln - labels

Labels are identifiers for each line of input file. They are used for addressing each line of file. The labels should be unique. Addressing lines of file means, the required start label entered by user defines line in input file from which serial values reading starts.

addr -

Addr defines buffer address to write data following the address.

byte0..byten, byte0..bytem, byte0..bytek -

Bytes arrays byte0..byten, byte0..bytem and byte0..bytek are defining data, which are assigned to write to buffer. Maximum count of bytes in one data field following the



address is 64 bytes. Data bytes are written to buffer from address `addr` to `addr+n`.

The process of writing particular bytes to buffer is:

```
byte0 to addr
byte1 to addr + 1
byte2 to addr + 2
....
byten to addr + n
```

Optional part is delimited from the first data part by character “ , ” (comma) and its structure is the same as in the first data part, i.e. address and following array of data bytes.

Characters with special use:

[] - labels must be defined inside square brackets

',' – character which delimiters basic part and optional part of data

';' - the semicolon character means the beginning of a comment. All characters from “;” to the end of line are ignored. Comment can be on individual line or in the end of definition line.

Note: *Label names can contain all characters except '[' and ']'. The label names are analysed as non case sensitive, i.e. character 'a' is same as 'A', 'b' is same as 'B' etc..*

All address and byte number values in input file are hexadecimal.

Allowed address value size is from 1 to 4 bytes.

Allowed size of data arrays in one line is in range from 1 to 64 bytes. When there are two data arrays in one line, the sum of their size in bytes can be maximally 80 bytes.

Be careful to set correct addresses. Address must be defined inside device start and device end address range. In case of address out of range, warning window appears and serialization is set to disabled (None).

Example:

```
[nav1] A7890 78 89 56 02 AB CD ; comment1
[nav2] A7890 02 02 04 06 08 0A
[nav3] A7890 08 09 0A 0B A0 C0 ; comment2
[nav4] A7890 68 87 50 02 0B 8D
[nav5] A7890 A8 88 59 02 AB 7D
```

;next line contains also second definition

```
[nav6] A7890 18 29 36 42 5B 6D , FFFF6 44 11 22 33 99
88 77 66 55 16
```



; this is last line - end of file

In the example file six serial values with labels „nav1“, „nav2“, ...“nav6“ are defined. Each value is written to buffer on address \$A7890. All values have size 6 bytes. The line with „nav6“ label has also second value definition, which is written to buffer on address \$FFFF6 and has size 10 bytes, i.e. the last byte of this value will be written to address \$FFFFF.

Device / Device options / Statistics

Statistics gives the information about actual count of device operations, which were proceeded on selected type device. If one device is corresponding to one device operation, e.g. programming, the number of device operations will be equal to number of programmed devices.

The next function of statistics is **Count down**. Count down allows checking the number of device operations, and then number of devices, on which device operations have to be done. After each successful device operation the value of count down counter is decremented. Count down has user defined start number of devices to do. When count down value reach zero, it means, specified number of devices is complete and user message about complete count down will be displayed.

Statistics dialog contains following options:

Check boxes **Program**, **Verify**, **Blank**, **Erase** and **Read** define operations, after which statistics values increment.

Check box **Count down** sets Count down activity (enable or disable). Edit box following the Count down check box defines initial number of count down counter, from which count down starts.

Statistics dialog can be also opened by pressing right mouse button on Statistics panel and clicking displayed item Statistics.

Actual statistics values are displaying in main window of control program in Statistics panel.

Statistics panel contains three statistics values – **Success**, **Failure**, **Total** and two Count down information values **Count down** and **Remains**.

Meaning of the values is:

Success number of operations which where successfully completed



Failure	number of operations which were not successfully completed
Total	number of all operations
Count down	informs about Count down activity (Enabled or Disabled)
Remains	informs about remaining number of device operations to do

Successful operation means any device operation of these, which is completed without errors:

- program
- verify
- blank check
- erase
- read

If device operation is finished with error(s) it is not successful operation.

When new device type is selected, all statistics values are set to zero and **Count down** is set to **Disabled**.

Reset button in **Statistics** panel reset statistics values.

Reload Count down button in **Statistics** panel reloads initial value to **Count down**.

Device / Device options / Associated file

This command is used for setting associated file with current device. This is a file, which can be automatically loaded to buffer after device is selected from default devices select list or by start control program.

You can edit the associated file name in file name box, put a full pathname. The control program checks the presence of this file on the disk. Also is possible enabling or disabling automatic load of this file.

You can save both settings i.e. associated file and enabling of automatic load of this file to disk by command **File / Exit and save**.

Device / Blank check

This command allows to blank check of all devices or its part if possible. The control program reports a result of this action by a write of a warning message to INFO window.

The menu command **Device / Device options / Operation options** allows to set another working area as the standard.



Device / Read

This command allows to read all device or its part into the buffer. The control program reports a finish of this action by write a message to INFO window.

The menu command **Device / Device options / Operation options** allows to set another working area as the standard. Setting an option **Verify data after reading** in this menu command means a higher reliability for device reading.

Device / Verify

This command compares the programmed data of the all device or its part with data in buffer. The control program reports a result of this action by a write of an error message to INFO window.

The menu command **Device / Device options / Operation options** allows to set another working area as the standard.

By the setting in the menu **Options / Display errors** the command lets to write the found errors on the display or write the found errors to VERIFY.ERR file. In the Display errors mode to the screen can display the program max. 45 the first found differences, which are located by the address where they were caused.

Device / Program

This command allows to programming of the all device or its part by the data of the buffer. The control program reports a result of this action by a write of an error message to INFO window.

The menu command **Device / Device options / Operation options** allows to set another working area as the standard, and set other operation options for programming process control.

Device / Erase

This command allows to erase the all programmable device. The program reports the end without error or end with the error by writes the warning report on the display.



Device / Test

This command executes a test with device selected from list of supported devices (e.g. static RAM) on programmers, which support this test.

Device / Device info

The command provides additional information about the current device - size of device, organization, programming algorithm and a list of programmers (including auxiliary modules), that supported this device. You can find here a package information and other general information about current device too.

The reserved key **<Ctrl+F1>** will bring out this menu from any menu and any time immediately.

Device / Device list

This command makes a list of all devices supported by current programmer and saves its to ?????DEV.TXT text file in the directory where control program is run from.



Buffer

This submenu is used for buffer manipulation, block operation, filling a part of buffer with string, erasing, checksum and of course editing and viewing with other items (find and replace string, printing...).

Buffer / View/Edit

This command is used to **view** (view mode) or **edit** (edit mode) data in buffer (for viewing in DUMP mode only). Use arrow keys for select the object for edit. Edited data are signified by colour.

You can use <F4> hot key also.

View/Edit Buffer

F1	display help of actual window
F2	fill block causes filling selected block of buffer by requested hex (or ASCII) string. Sets start and end block for filling and requested hex or ASCII string.
Ctrl+F2	erase buffer with specified blank value
Ctrl+Shift+F2	fill buffer with random data
F3	copy block is used to copy specified block of data in current buffer on new address. Target address needn't be out from source block addresses.
F4	move block is used to move specified block of data in current buffer on new address. Target address needn't be out from source block addresses. Source address block (or part) will be filled by topical blank character.
F5	swap bytes command swaps a high- and low- order of byte pairs in current buffer block. This block must started on even address and must have an even number of bytes. If this conditions do not fulfil, the program modifies addresses itself (start address is moved on lower even address and/or end address is moved on higher odd address).
F6	print buffer
F7	find string (max. length 16 ASCII characters)
F8	find and replace string (max. 16 ASCII chars.)
F9	change current address
F10	change mode view / edit



F11	switch the mode of buffer data view between 8 bit and 16 bit view. It can be also do by mouse clicking on the button to the right of View/Edit mode buffer indicator. This button indicates actual data view mode (8 bit or 16 bit), too.
F12	checksum dialog allows to count checksum of selected block of buffer change mode view / edit
Arrow keys	move cursor up, down, right and left
Home/End	jump on start / end current line
PgUp/PgDn	jump on previous / next page
Ctrl+PgUp/PgDn	jump on start / end current page
Ctrl+Home/End	jump on start / end current device
Backspace	move cursor one position left (back)

Note: *characters 20H - FFH (mode ASCII) and numbers 0..9, A..F (mode HEX) immediately changes content of edit area.*

Warning: Editing of ASCII characters for word devices is disabled.

Print buffer

This command allows write selected part of buffer to printer or to file. Program uses at it an external text editor in which selected block of buffer is displayed and can be printed or saved to file, too. By default is set simple text editor **Notepad.EXE**, which is standard part of all versions of MS Windows.

In Print buffer dialog are following options:

Block start

Defines start address of selected block in buffer.

Block end

Defines end address of selected block in buffer.

External editor

Defines path and name of external program, which has to be used as text viewer for selected block of buffer. By default is set simple text editor NOTEPAD.EXE, which is standard part of all versions of MS Windows. User can define any text editor for example WORDPAD.EXE, which is able to work with large text files. In user defined text editor user can print or save to file selected block of buffer. The external editor path and name is saved automatically to disk.

Find text dialog box

Enter the search string to **Find** to text input box and choose **<OK>** to begin the search or choose **<Cancel>** to forget it.



Direction box specifies which way you want to search, starting from the current cursor position (In edit mode).

Forward (from the current position or start of buffer to the end of the buffer) is the default. **Backward** searches toward the beginning. In view mode searches all buffer.

Origin specifies where the search should start.

Replace text dialog box

Enter the search string in the **Text to find** string input box and enter the replacement string in the **Replace with** input box.

In **Options** box you can select prompt on replace: if program finds instance you will be asked before program change it.

Origin specifies where the search should start.

Direction box specifies which way you want to search, starting from the current cursor position (In edit mode).

Forward (from the current position or start of buffer to the end of the buffer) is the default. **Backward** searches toward the beginning. In view mode searches all buffer.

Press <Esc> or click **Cancel** button to close dialog window.

By pressing **Replace** button the dialog box is closed and a Question window is displayed. This window contains following choices:

Yes replaces found item and finds next

No finds next item without replacing current one

Replace All replaces all found items

Abort search aborts this command

View/Edit buffer for PLD

Ctrl+F2 erase buffer with specified blank value

Ctrl+Shift+F2 fill buffer with random data

F9 goto address...

F10 change mode view / edit

F11 switch the mode of buffer data view between 1 bit and 8 bit view. It can be also do by mouse clicking on the button to the right of View/Edit mode buffer indicator. This button indicates actual data view mode (1 bit or 8 bit), too.

Arrow keys move cursor up, down, right and left

Home/End jump on start / end current line

PgUp/PgDn jump on previous / next page

Ctrl+PgUp/PgDn jump on start / end current page

Ctrl+Home/End jump on start / end edit area

Backspace move cursor one position left (back)



Note: Characters 0 and 1 immediately changes content of edit area.

Buffer / Fill block

Selecting this command causes filling selected block of buffer by requested hex (or ASCII) string. Sets start and end block for filling and requested hex or ASCII string.

Buffer / Copy block

This command is used to copy specified block of data in current buffer on new address. Target address needn't be out from source block addresses.

Buffer / Move block

This command is used to move specified block of data in current buffer on new address. Target address needn't be out from source block addresses. Source address block (or part) will be filled by topical blank character.

Buffer / Swap block

This command swaps a high- and low- order of byte pairs in current buffer block. This block must started on even address and must have an even number of bytes. If this conditions do not fulfil, the program modifies addresses itself (start address is moved on lower even address and/or end address is moved on higher odd address).

Buffer / Erase

If this command is selected, the content of the buffer will be filled with topical blank character.

The reserved key **<Ctrl+F2>** will bring out this menu from any menu and any time.

Buffer / Fill random data

If this command is selected, the content of the buffer will be filled with random data.

The reserved key **<Shift+Ctrl+F2>** will bring out this menu from any menu and any time.

Buffer / Checksum

The checksum dialog is used for calculate checksums of selected block in buffer. The checksums are calculated by next way :



Byte	sum by bytes to "word". CY flag is ignored
Word	sum by words to "word". CY flag is ignored
Byte (CY)	sum by bytes to "word". CY flag is added to result.
Word (CY)	sum by words to "word". CY flag is added to result.
CRC-CCITT	sum by bytes to "word" using $RESULT = PREVIOUS + (x^{16} + x^{12} + x^5 + 1)$
CRC-XModem	sum by bytes to "word" using $RESULT = PREVIOUS + (x^{16} + x^{15} + x^2 + 1)$

Column marked as **Neg.** is a negation of checksum so, that
 $Sum + Neg. = FFFFH$.

Column marked as **Suppl.** is complement of checksum so,
that $Sum + Suppl. = 0$ (+ carry).

Dialog checksum contains following items:

From address: This is a start address of block selected for calculating checksums in buffer. Address is defined as Byte address.

To address: This is an end address of block selected for calculating checksums in buffer. Address is defined as Byte address.

Insert checksum: This is special item used for select which kind of checksum will be written into the buffer when, the **Calculate & insert** was executed.

Insert address: This is special item that specifies an address from the buffer where a result of chosen checksum will be written, when the **Calculate & insert** was executed. Address can not be specified inside the range **<From address>** to **<To address>**. Address is defined as Byte address.

Size: This item is used for setting a size of chosen checksum result, which will be written into the buffer. A size of checksum result may be 8 (byte) or 16 (word) bits long. If word size was selected, whole checksum value will be written into the buffer. In other case will be written only low byte of checksum value.

Note: *If word size was selected, a low byte of checksum value will be written on address specified in box Insert address and a high byte will be written on address incremented by one.*



Calculate: Click on the button Calculate starts calculating checksums for selected block in buffer. No writes into the buffer are executed.

Calculate & insert: Click on the button **Calculate & insert** starts calculating checksums for selected block in the buffer and writes the chosen checksum into the buffer on address specified by **Insert address**.



Options

The Options menu contains commands that let you view and change various default settings.

Options / General options

General options dialog allows user to control following options of program.

Buffer

This page contains several options for loading control by any of HEX formats.

Erase buffer before loading.

This option sets erasing buffer (with desired value) automatically before the loading by any of HEX formats.

Negative offset before loading.

This option sets a negative offset, which is used for data addresses modification by loading from any HEX file so, that data can be written to existing buffer addresses.

Example:

A file contains data by Motorola S - format. A data block started at address FFFF0H. It is a S2 format with length of address array of 3 bytes. For all data reading you can set a value of negative offset to FFFF0H. It means, that the offset will be subtracted from current real addresses and so data will be written from buffer address 0.

Warning: *The value of negative offset is subtracted from real address and therefore a result of subtraction can be negative number. Because take care of correct setting of this value.*

Intel hex loading.

This option sets data redirecting from file by extended **Intel HEX** format to physically existing buffer addresses. It means, that the user will be prompted to enter segment, which is associated with the lowest segment in file (record type 02) and the other segments will be modified with this difference. This option isn't valid for "simple" Intel HEX format. Default set means inactive status.

Example:

The file contains two records of type 02 with addresses F000H and F800H. When we enter new segment for example 0H, data from segment F000H will be directed to segment 0H and similar data from segment F800H to segment 800H.



File options

File options page allows you to choose loading file format recognizing and set file masks.

Load file format

Use this command for set file-name masks to use as a filter for file listing in **File / Save** and **File / Load** file window for all file formats. Mask must contain one of wildcards (*, ?) at least and must be applied correctly by syntax.

When **automatic** file format is selected, program analyses format of loading file and test file for each of supported formats that are available in program. If file format matches one of supported formats, the file is read to buffer in detected format.

Manual file format allows user to select explicitly wished file format from list of supported file formats. File may be loaded incompletely or incorrectly, if file format does not match to user selected format.

File format masks

Use this command for set file-name masks to use as a filter for file listing in **File / Save** and **File / Load** file window for all file formats. Mask must contain one of wildcards (*, ?) at least and must be applied correctly by syntax.

Language

This page allows you to select another language for user interface such as menu, buttons, dialogs, information and messages. It also allows to select wished help file in another language. For another language support of user interface the language definition file is required.

Sound

Sound page allows user to select the sound mode of program. Program generates sounds after some activities, e.g. activities on device (programming, verifying, reading, etc.). Program generates sound also when warning or error message is displayed. User can now select sound from MS Windows system sound (required installed sound card), PC speaker or none sound.

Other

Page Other allows user to manage other program settings, e.g. hint display options on toolbar buttons in main program window.

Save options

Save options page allows you to select the program options saving when exiting program. Three options are available here:



Don't save options - don't save options during quitting program and don't ask for saving options.

Auto save options - save options during quitting program without asking for saving options.

Prompt for save options - program asks user for saving options before quitting program. User can select to save or not to save options.

Options / View

Use the View menu commands to display or hide different elements of program environment such as toolbars.

Options / View / Main toolbar

Choose this command to show or hide the Main toolbar.

Options / View / Additional toolbar

Choose this command to show or hide the Additional toolbar.

Options / Display errors

This option allows you set a form of errors displaying as a result of programmed data verifying. Errors can be displayed to the **screen** (max. 45 differences), **saved to VERIFY.ERR** file on the disk in current directory or it will **not displayed**. In case the displaying errors are turned off, the control program reports a warning message in INFO window only.

This setting can be saved to disk by command **Options / Save options**. Default form is set to a screen displaying.

Options / Find programmer

Selects a new type of programmer and communication parameters. This command contains following items:

Programmer - sets a new type of programmer for find. If a Search all is selected, the control program finds all supported programmers.

Establish communication - allows manual or automatic establishing communication for a new programmer.

Speed - sets speed, if a manual establishing communication is selected, which PC sends data into the programmer. Speed is expressed as a percent from a maximal speed.

The communication speed modification is important for PCs with "slow" LPT ports, which haven't sufficient driving power for a PC<->programmer cable (laptop, notebook, ...). Use this command, if you have any communication problems with connected programmer on the LPT port of your PC (e.g. control program reports a programmer



absence, the communication with the programmer is unreliable, etc.).

If automatic establishing communication is selected, then control program sets a maximal communication speed.

Port - selects a LPT port, which will be scanned for a requested programmer. If All port is selected, the control program scans all LPT ports, which are available on standard addresses.

Address for special port - sets address of LPT port, if a Special port is selected.

Pressing key **<Enter>** or button **OK** initiates scanning for programmer by set parameters. There is same activity as at start the control program. The command clears a list of default devices without the current device, if the new selected programmer supports this one.

This setting is saved to disk by command **Options / Save options**.

Options / Automatic YES!

This command is used for setting **Automatic YES!** mode. In this mode you just put a device into ZIF socket and a last operation will be repeated automatically. Program automatically detects an insertion of a new device and runs last executed operation without pressing any key or button. An insertion of device into ZIF is displayed on the screen. Repeated operation executing will be cancelled by pressing key **<Esc>** during waiting for insert/remove a device to/from ZIF.

Note: *During waiting for an insertion a new device into ZIF socket, the LED BUSY on the programmer is blinking.*

This mode may be enabled or disabled by item **Automatic YES!** mode. If a new programmer is selected **Options / Find programmer**, this mode will be disabled.

In **Response time** is possible to set a time interval within must be detected device in ZIF socket to accept an insertion of a new device. Default is set standard interval. If socket adapter is used then is recommended to set an elongated interval.

In **Pins with capacitors** bar may be entered a list of a pins interconnected by capacitors (for example: if a converter, which have connected capacitor between VCC and GND,



is used), which may makes problems at detecting insertion of a new device.

List of pins of device is in form:

pinA, pinB, pinC....

Example: 4,6,17

This list is erased if a new device is selected by **Device / Select default** or **Device / Select device ...**

This setting is saved to disk by command **Options / Save options**.

Options / Log file

This options associates with using of **Log window**. All reports for this window can be written into the **Log file** too. The Log file name as **REPORT.REP** and the control program creates this file in current directory.

Sets **New** caused deleting old Log file if exist and creating a new file for reports. Sets **Append** adds all reports into existing Log file. If file not exist, the new file will be created. Settings are applied only at program start.

This setting can be saved to disk by command **Options / Save options**. Default form is set to a using Log window without Log file i.e. all reports will be displayed to a Log window only.

Options / Save options

This command saves all settings that are currently supported for saving, even if auto-save is turned off. Following options are saved: options under the Options menu, ten last selected devices, file history, main program window position and size.



Diagnostics

This command includes selftest for programmers and IC test.

Diagnostics / Selftest

Command executes a selftest of current programmer without diagnostic POD. We recommend execute also **Diagnostics / Selftest plus** of programmer.

Diagnostics / Selftest plus

Command executes a selftest of current programmer using diagnostic POD, which is included in standard delivery of programmer. We recommend run this test as often as possible, e.g. once per month.

Diagnostics / IC test

This command activates a test section for ICs separated by compatibility to any libraries (on distribution diskette). First select an appropriate library, wished device and then a mode for test vectors run (**Loop, Single step**). Control sequence and test results are displayed to Log window.



Help

Pressing the <F1> key accesses the Help. When you selecting menu item and press <F1>, you access context-sensitive help. If PG4U / PG4UW is executing an operation with the programmer <F1> generates no response.

The following Help items are highlighted:

- words describing the keys referred to by the current Help
- all other significant words
- current cross-references; click on this cross-reference to obtain further information.

Since the Help system is continuously updated together with the control program, it may contain information not included in this manual.

Detailed information on individual menu commands can be found in the integrated on-line Help.

Note: *Updates to this manual should be read in the README_P.TXT file in the enclosed diskette*



Common notes



Software

PG4U/PG4UW is common control program for all of the ELNEC programmers. Thus, during work with him its possible to find some items, those refer not to current selected programmer.

There is possible to attach to selected file "device associated file" too. This file may be automatically loaded into the buffer, if item **Device options/Associated file/Automatic load** is set to Yes. File can be associated manually or accepting last loaded file into the buffer. For this purpose is used item **Device options/Associated file/Edit name**.

There exist feature of control program, named **Automatic YES!**. In concrete, it mean there isn't necessary to press "YES!" labeled button to repeat last activity, because auto-sensing feature of programmer detect insertion of new chip into ZIF socket and start last executed automatically. This feature can be set in menu Options. If you use package converter with come pins coupled by capacitor, enter list of these pins. **Automatic YES!** feature is available for JetProg, LabProg+ programmer only.

Some special devices (e.g. Philips Coolrunner family) require external DAT files, that isn't present in standard PG4U/PG4UW SW delivery on diskettes. If you need to program these devices, look at **www.elnec.com**, section Download.



Hardware

Due a large variety of parallel port types, a case may occur when the programmer cannot "get concerted" with the PC. This problem may be shown as none communication between the PC and the programmer, or by unreliable communication. If this behavior occur, try to connect your programmer to some other PCs or other parallel ports near you.

If you find none solution, please document the situation, i.e., provide us an accurate description of your PC configuration, including some other circumstances bearing on the problem in question, and advise the manufacturer of your problem. Don't forget please enter of PC type, manufacturer, speed, operation system, resident programs; your parallel port I/O manufacturer and type. Use please **Programmer problem form** for this purpose (see **Appendix A**).



Other

Attention to multitasking OS's (WIN 3.11/95/98/NT/2000). There is needful for regular running of control program for any ELNEC programmer, that printer port, on which is programmer connected, must be reserved for this programmer only. Otherwise, any other program must not simultaneously to use (or any way to modify) this printer port.

PG4U/PG4UW SW can handle all modes of LPT port (full IEEE 1284 support), thus you don't need to configure LPT port for connection of ELNEC programmers.

WIN98 have bug (or wittingly) in the MSDOS.SYS file. Initial setting of variable DoubleBuffer is 1, therefore DOS applications run slowly. Write please DoubleBuffer=0.

In case of WIN software, please don't move any window during BUSY LED is on - watching circuit can be activate to switch the programmer in safe status as in case communication PC-programmer error.



Troubleshooting and warranty



Throubleshooting

We really want you to enjoy our product. Nevertheless, problems can occur. In such cases please follow the instructions below.

- It might be your mistake in properly operating the programmer or its control program PG4U/PG4UW.
 - Please read carefully all the enclosed documentation again. Probably you will find the needed answer right away.
 - Try to install programmer and PG4U/PG4UW on another computer. If your system works normally on the other computer you might have a problem with the first one PC. Compare differences between both computers.
 - Ask your in-house guru (every office has one!).
 - Ask the person who already installed programmer.
- If the problem persists, please call the local dealer, from whom you purchased the programmer, or call ELNEC direct. Most problems can be solved by phone, e-mail or fax. If you want to contact us by:
 - **Mail/fax** - Copy the "**PROGRAMMER PROBLEM REPORT**" form and fill it in following the instructions at the end of the form. Write everything down that you consider to be relevant about the programmer, software and the target device. Send the completed form by mail or fax to ELNEC (fax number in the control program, submenu Info) or to your local dealer. If you send the form by fax please use black ink, a good pen and large letters!
 - **E-mail** - Use "**PROGRAMMER PROBLEM REPORT**" form on the enclosed diskette or from our Internet site and fill it in following the instructions at the end of the form. Use standard ASCII editor. Write everything down that you consider to be relevant about the programmer, software and the target device. Send the completed form by e-mail to your local dealer or to ELNEC (elnec@elnec.com).
 - **Phone** - Copy "**PROGRAMMER PROBLEM REPORT**" form and fill it in following the instructions at the end of the form. Write everything down that you consider to be relevant about the programmer, software and the target device. Send the completed form by mail or fax to ELNEC (fax number in the control program, submenu Info) or to your local dealer. If you send the form by fax please use black ink, a good pen and large letters easily to read. Then call your local dealer or ELNEC's customer support center (phone number in PG4U, submenu **Info**, in PG4UW **Help / About**). Please keep your manual, the programmer and the completed "**PROGRAMMER PROBLEM REPORT**" form (just faxed) available, so that you can respond quickly to our questions.



- If your programmer is diagnosed as defective, consult your local dealer or ELNEC about the pertinent repair center in your country. Please carefully include the following items in the package:
 - defective product
 - completed "**PROGRAMMER PROBLEM REPORT**" form
 - photocopy of a dated proof of purchase

Without all these items we cannot admit your programmer to repair.

Note:

You may find the "**PROGRAMMER PROBLEM REPORT**" form:

- in **Appendix A** of this manual
- on the enclosed floppy disk (file **PROPROR.TXT**)
- at our Internet site (**www.elnec.com**), section 'Notes'.
- if the period between purchasing the programmer and the defective incident is greater than 6 months please also include the "**Extended warranty**" document with your own registration number. For more information please read the chapter "**Warranty terms**".



If you have an unsupported target device

If you need to operate on a target device not supported by the control program for programmer, please do not despair and follow the next steps:

- Look in the device list of the latest version of the control program on our Internet site (section Download, file corresponded to your programmer). Your new target device might already be included in this version! If yes, download the file (for DOS) or PG4UWARC.EXE (for WIN) and install the new version of the control program.
- Contact ELNEC direct, filling up a "**Device Problem Report**" form following the instructions at the end of this form. We may need detailed data sheets of your target device and, if possible, samples. The samples will be returned to you after we include your target device in a new version of PG4U/PG4UW.

Note:

See also AlgOR service in Appendix D in this manual.

*You may find the "**Device Problem Report**" form:*

- *in **Appendix B** of this manual*
- *on the enclosed floppy disk (file **DEVPROR.TXT**)*
- *at our Internet site (www.elnec.com), section 'Notes'.*



Warranty terms

The manufacturer, ELNEC s.r.o. Presov, Slovakia, gives a guarantee on failure-free operating of the programmer and all its parts, materials and workmanship for **one-year** (51&AVRprog, PIKprog and SEEprog) or **three-year** (JetProg, LabProg+, PREPROM-02aLV) from the date of purchase. This warranty is limited to 25,000-cycles on DIL ZIF socket or 10,000-cycles on PLCC ZIF sockets). If the product is diagnosed as defective, ELNEC s.r.o. or the authorized repair center will repair or replace defective parts at no charge. Parts used for replacement and/or whole programmer are warranted only for the reminder of the original warranty period.

For repair within the warranty period, the customer must prove the date of purchase. Extended warranty is available only for customers who have the "**Extended warranty**" document with their own registration number from ELNEC (please read note at the end of this chapter). All other programmer owners only carry a standard 6 months warranty.

The warranty does not apply to products that are of wear and tear or mechanically damaged. Equally, the warranty does not apply to products opened and/or repaired and/or altered by personnel not authorized by ELNEC, or to products that have been misused, abused, accidentated or that were improperly installed.

For unwarrantable repairs you will be billed according to the costs of replacement materials, service time and freight. ELNEC or its distributors will determine whether the defective product should be repaired or replaced and judge whether or not the warranty applies.

Note:

- Every customer obtains an "**Extended warranty**" document and an own registration number if he/she sends to ELNEC the filled out **registration** form within 4 weeks from the purchase of the programmer. The Registration form is part of the standard programmer package. If this form is missing, please copy it from **Appendix E** of this manual.
- Please also see Troubleshooting section.

Manufacturer:

✉: **ELNEC s. r. o.**, Post 5, P. O. Box 22, SK - 08005 Presov, Slovakia
☎: +42191/77 34 328, 77 31 007, fax 77 32 797
www.elnec.com, elnec@elnec.com



Appendix



Appendix A - Programmer Problem Report form

Please make a copy of this page to A4 or use the A4 form enclosed in standard delivery package. This form can also be found on the enclosed floppy disk and on our Internet site.

PROGRAMMER PROBLEM REPORT

	<i>Example</i>	
To (if you know name of contact person)	<i>Ing. Puobiš</i>	_____
Date	<i>jan 19, 2001</i>	_____
Subject (title of case)	<i>LabProg+</i>	_____
Customer		
Company	<i>Moovie Studio</i>	_____
Name	<i>John Siboup</i>	_____
Address	<i>Tom 6, Praha</i>	_____
Distributor	<i>CENLE, Wien</i>	_____
Voucher number	<i>enclosed copy</i>	_____
Date of purchase	<i>jan 16, 2000</i>	_____
Data of claiming	<i>jan 31, 2000</i>	_____
Information about product		
Name of product	<i>LabProg+</i>	_____
Serial number	<i>050954</i>	_____
Control program	<i>PG4UW.EXE</i>	_____
Version	<i>1.24</i>	_____
Information about PC		
Processor, speed	<i>P3, 800</i>	_____
Version of OS	<i>WIN ME</i>	_____
Location of LPT port	<i>add-on card</i>	_____
	<i>COMBO</i>	_____
Type and setting of LPT port	<i>EPP/ECp</i>	_____
	<i>/ set as SPP</i>	_____
Information about environment		
Surroundings	<i>production line</i>	_____
Temperature	<i>-10 to 35 °C</i>	_____

Below description should included

step-by-step description of all activities that invokes a problem

Please make your problem description as specific as possible - you can increase speed and chance to resolve a problem. This description should be included either what is known to word correctly, either what is failing. Copy all error reports too. Yours comments and descriptions of expectations are welcomed.

Note Before you send product to repair, answer self to some questions:

- Does programmer work well with others PC or under others operating systems?
- Are you using the programmer in accordance with technical specification of programmer and with original accessories?
- Isn't ZIF socket of programmer or adapter dusty or out of life?
- Does programmer work successfully with other devices?
- Does problems occurred steadily or occasionally?



Appendix B - Device Problem Report form

Please make a copy of this page to A4 or use the A4 form enclosed in standard delivery package. This form can also be found on the enclosed floppy disk and on our Internet site.

DEVICE PROBLEM REPORT

Example

To (if you know name of contact person) **Ing. Puobiš** _____
Date **jan 19, 2001** _____
Subject (title of case) **LabProg+** _____

Customer

Company **Moovie Studio** _____
Name **John Siboup** _____
Address **Tom 6, Praha** _____
Distributor **CENLE, Wien** _____
Voucher number **enclosed copy** _____
Date of purchase **jan 16, 2000** _____
Data of claiming **jan 31, 2000** _____

Information about product

Name of product **LabProg+** _____
Serial number **050954** _____
Control program **PG4UW.EXE** _____
Version **1.24** _____

Information about device

Device type **AT24C08** _____
Manufacturer /logo **Atmel** _____
Origin **Philippines** _____
Date code **9742** _____
All other numbers on device **top: 12** _____
1bottom: 233039 _____
Package **DIL8** _____

Below description should included

step-by-step description of all activities that invokes a problem

Please make your problem description as specific as possible - you can increase speed and chance to resolve a problem. This description should be included either what is known to word correctly, either what is failing. Copy all error reports too. Yours comments and descriptions of expectations are welcomed. It's best, send to us device with which problem occurs.

Note Before you send above mentioned form, answer self to some questions.

- Does programmer work well with others PC or under others operating systems? ZIF socket of programmer or adapter isn't dusty or out of life?
- Is the device new or used? Is error reported for all of the tested device or only for one? Is error reported for devices with other date code?
- Does programmer work successfully with other devices?
- Does programmer work successfully with other devices?



Appendix C - Keep-Current Service

Keep-Current is a subscription service, with which your ELNEC programmer and documentation is kept up-to-date with the latest device support list and with the last version of the control program and its new features.

ELNEC ships off the latest version of the programmer software and updated user documentation (Keep-Current package). A Keep-Current service is your task-free guarantee that you are exploiting your ELNEC programmers and doing high quality programming at maximum and with minimal costs.

Keep-Current package can be mailed to you in two ways:

- by standard post
- by e-mail.

According to your needs you may choose three levels of Keep-Current service:

- **QUICK** ☞ Keep-Current package is mailed to you immediately after each relevant change of software or documentation (approximately monthly). This service carries a fee (please consult price list) and it's available only by e-mail.
- **STANDARD** ☞ Keep-Current package is mailed to you quarterly. This service again carries a cost, (please consult price list) and is available by post or by e-mail.
- **1/Y** ☞ Keep-Current package is mailed to you once per year. This service is available only by e-mail and is free - but a registration is required.

Keep-Current package (file KC_ARC.EXE) contains:

- latest version of the control program (file PG4UARC.EXE or PG4UWARC.EXE)
- latest device list
- description of added/removed device and/or modified algorithms
- description of added and/or modified features of the control program
- latest information about ELNEC's products

<i>Prices of Keep-Current service (P&P included)</i>		
<i>level</i>	<i>e-mail</i>	<i>per post</i>
QUICK	50DM / 30USD	not offered
STANDARD	25DM / 15USD	60DM / 35USD
1/Y	free	postage / packing cost



Why is it important to use the latest version of the control program?

- Semiconductor manufacturers continuously introduce new devices with new package types, manufactured by new technologies in order to support the need for flexibility, quality and speed in product design and manufacturing. To keep pace and to keep you up-to-date, we usually implement more than 500 new devices into the control program within a year.
- Furthermore, a typical programmable device undergoes several changes during its lifetime in an effort to maintain or to improve its technical characteristics and process yields. These changes often impact with the programming algorithms, which need to be upgraded (the programming algorithm is a set of instructions that tells the programmer how to program data into a particular target device). Using the newest algorithms in the programming process is the key to obtaining high quality results. In many cases, while the older algorithm will still program the device, they may not provide the level of data retention that would be possible with an optimal algorithm. Failure to not use the most current algorithm can decrease your programming yields (more improper programmed target devices), and may often increase programming times, or even affect the long term reliability of the programmed device.
- At least, we are making mistakes too

Our commitment is to implement support for these new or modified parts before or as soon as possible after their release, so that you can be sure that you are using latest and/or optimal programming algorithms that were created for this new device.



Appendix D - AlgOR service

(Algorithms On Request)

AlgOR is a free service, by which we respond, as flexible as possible on the customer's request to implement programming support for new devices. This service may be used also for requesting new features of the control program.

AlgOR process is simple. The user sends to ELNEC a request for additional support for XXX device to the control program (we may ask for up-to-date data sheets and samples, if needed). After completion, the user will obtain a new version of the control program with requested features. We will, of course, also return the borrowed samples. If we cannot satisfy your requirements (too expensive, algorithms not available, additionally module needed), we will promptly contact to you and propose an appropriate solution.

Note:

Please use "AlgOR (Algorithms On Request)" form and send it direct to ELNEC.

*AlgOR service is **free** of charge. Therefore we do not accept any claims regarding this service. ELNEC s.r.o. reserves the right to set the dispatching priority on the particular tasks according to its own judgement.*



Appendix E - registration card

If the registration card is missing from your standard programmer delivery package please use a copy of the form below and send it to ELNEC. We remind you that without the "Extended warranty" document only the standard 6-month warranty is granted.

✂-----

Device: LabProg+
Serial number: _____
Name: _____
Company: _____
Department: _____
Address: _____
Post Code, city: _____
Country: _____
phone/fax/e-mail: _____
Date of purchase: _____
Purchase from: _____

Complete, please:

- what type of computer are you using:
☐AT ☐386 ☐486 ☐Pentium ☐better
- operating system: ☐MS-DOS ☐DR-DOS,
☐WINDOWS 3.x ☐WINDOWS 95/98 ☐WINDOWS
NT/2000 ☐OS/2
- how did you first hear about ELNEC ?
☐advert. ☐dealer ☐other
- why did you choose this product?
☐price ☐quality ☐recommendation
☐features (please specify) _____
- comments: _____

**F/2 - CE certificate PREPROM-02aLV**

EC Attestation of Conformity

No.: E8 99 07 36878 001

TÜV
PRODUCT SERVICE

ELNEC, spol. s r.o.
Petrovská 38, Praha 6, P.O. Box 22
SK - 080 05 PREŠOV

Item Identification: Information technology equipment (EMC directive)
Universal EPROM programmer
PREPROM-02aLV

Item Description: 12V DC
8W

The equipment complies with the principal protection requirements of the EMC Directive (Directive 89/336/EEC relating to electromagnetic compatibility) based on a voluntary test.

This attestation applies only to the particular sample of the product and its technical documentation provided for testing and certification. The detailed test results and all standards used as well as the operation mode are listed in

Test report no.: E 016L KO

Test standards: EN 55022/1994, EN 55022/A1:1996, EN 55022/A2:1997, EN 55024/1998

This attestation is released with the above mentioned attestation number by the Certification Body of TÜV PRODUCT SERVICE. It does not permit the use of a TÜV PRODUCT SERVICE certification mark.

After preparation of the necessary technical documentation as well as the conformity declaration the CE marking as shown below can be affixed on the equipment as stipulated in Article 10.1 of the Directive. Other relevant Directives have to be observed.

CE 

Department: EMC / STK / KO
Date: July 22, 1999

TÜV
PRODUCT SERVICE GMBH

Zertifizierungsstelle · Rinderstrasse 31 · D-80335 München

ZERTIFIKAT + CERTIFICATE + CERTIFICADO + CERTIFICAT + СЕРТИФИКАТ + CERTIFICATE



F/3 - CE certificate 51&AVRprog

Military Technical Institute of Ground Forces
V. Nejedlého 691, 682 03 Vytkov
Czech Republic

CERTIFICATE
No. VTÚPV - 009 / 2000

Applicant: ELNEC s.r.o.
Lomnická 2, 080 05 Prešov, Slovakia

Product: **Programmer**

Type: **51&AVRprog**

Manufacturer: ELNEC s.r.o.
Lomnická 2, 080 05 Prešov, Slovakia

Rating and principal characteristics: DC input 12 V/0,5 A; power input 3 W.

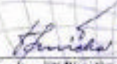
Test results are described
in the Test Reports No.: 730-401/2000, 730-407/2000.

The sample of the tested product conforms with the following standards:
- ČSN EN 55022B/1996, ČSN EN 51000-4-2/1997, ČSN EN 61009-4-3/1997.

This certificate is valid until: **30.06.2003**

Vytkov 06.05.2000




Ing. Jiří Pláněka
Head of the Certification Group



F/4 - CE certificate PIKprog

Military Technical Institute of Ground Forces
V. Nejedlicho 691, 682 03 Vyškov
Czech Republic

CERTIFICATE
No. VTÚPV - 008 / 2000

Applicant: **ELNEC s.r.o.**
Lomnická 2, 080 05 Prešov, Slovakia

Product: **Programmer**

Type: **PIKprog**

Manufacturer: **ELNEC s.r.o.**
Lomnická 2, 080 05 Prešov, Slovakia

Rating and principal characteristics: DC input 12 V/0,5 A; power input 3 W.

Test results are described
in the Test Reports No.: 710-399/2000, 710-404/2000.

The sample of the tested product conforms with the following standards:
- ČSN EN 55022B/1996, ČSN EN 61000-4-2/1997, ČSN EN 61000-4-3/1997.

This certificate is valid until: **30.06.2003**

Vyškov 06.06.2000




Ing. Jiří Pláněka
Head of the Certification Group



F/5 - CE certificate SEEprog

Military Technical Institute of Ground Forces
V. Nejedlého 691, 682 03 Vytkov,
Czech Republic

CERTIFICATE
No. VTÚPV - 010 / 2000

Applicant: **ELNEC s.r.o.**
Lomnická 2, 080 05 Prešov, Slovakia

Product: **Programmer**

Type: **SEEprog**

Manufacturer: **ELNEC s.r.o.**
Lomnická 2, 080 05 Prešov, Slovakia

Rating and principal characteristics: DC input 12 V/0,5 A; power input 3 W.

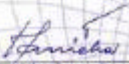
Test results are described
at the Test Reports No.: 730-405/2000, 730-405/2000.

The sample of the tested product conforms with the following standards:
- ČSN EN 55022B/1996, ČSN EN 61000-4-2/1997, ČSN EN 61000-4-3/1997.

This certificate is valid until: **30.06.2003**

Vytkov, 06.06.2000




Ing. Jiří Planíčka
Head of the Certification Group